2022 International Symposium on Control Engineering and Robotics (ISCER) | 978-1-6654-8478-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCER55570.2022.00015

Mobile Robot Path Planning in 2D space: A survey

Zishu Chai[1, *, †]
Logistics engineering college
Shanghai maritime university
Shanghai, 201306, China
iec@shmtu.edu.cn
Shanghai, China
*chai0005@hz.nl

Zongyuan Zhang[2, †]
Mechanical electrical engineering school
Beijing information science & technology university
Beijing, 100096, China
2018010144@mail.bistu.edu.cn
Beijing, China
15011088784@163.com
†These authors contributed equally.

***Abstract*—For decades, the path planning method has evolved at a fast speed ever and in a great amount. This paper briefly summarizes the existing path planning algorithm of mobile robots in a 2D plane. We classify path planning algorithms into four categories, namely classical methods, bionic methods, artificial intelligence methods, and hybrids methods, and introduce representative algorithms of each category respectively. A thorough review and classification of existing path planning algorithms are provided, which is beneficial for beginners in mobile robotics research. We demonstrate principal ideas for each type of path planning algorithm and provide discussions of the feasibility of real-world application for practitioners.**

Keywords—mobile robots, path planning, heuristic algorithms, bio-inspired algorithms, artificial intelligence

I. INTRODUCTION

A. Review Motivation

Over the past few decades, various advances in automation technologies have raised our quality of life to a new level. For example, sweeping robots can automatically scan the terrain, avoid obstacles and then clean the ground. In the whole process of wiping the floor, algorithms concerning path planning play an important role. By definition, path planning is an algorithm that plans the shortest and collision-free path for a robot to reach a given target position. This paper mainly discusses two-dimensional path planning algorithms.

At present, there are many kinds of classical path planning algorithms. Because each kind of algorithm adopts completely different ideas to find a path, algorithms need diverse preconditions when they can be used correctly. And there is also a difference between the success rate of finding a feasible path and whether it can find the shortest path. Classical path planning algorithms appeared as early as the 1960s, and some of them have been improved or reconstructed by researchers in recent years. In the 60 years since then, some new path planning algorithms or ideas have also appeared. In general, although some new path planning algorithms have not been applied by an entity project, most of the new path planning algorithms have been theoretically analyzed and experimentally verified. These verified new path planning algorithms have advantages over traditional path planning algorithms in robustness, accuracy, or real-time. Therefore, in recent years, path planning technology has been improved, and great progress has been made to an advanced degree. Because there are many kinds of path planning algorithms, researchers who are not familiar with the field of path planning may encounter difficulties in algorithm selection when they need to use path planning. At the same time, some development trends of path planning algorithms can be summarized from the upgrade of algorithms in recent years. Therefore, this paper classifies and summarizes different path planning methods in chronological order, reports distinct features by comparison table, and envision possible prospects based on mainstream developments.

B. Background and current situation

Due to the important applications of path planning in real-world scenarios, the provision of high-performance and realistic paths attracts a great deal of research. Over the decades, basic concepts such as A* [1], rapidly-exploring random tree (RRT) [2], artificial potential field(APF) [3], ant colony [4], genetic algorithm(GA) [5], fuzzy logic control [6], and neural network [7] have attracted many innovative variants, and hybrids of them have also been proposed recently to take advantage of core ideas in specific classic algorithms.

978-1-6654-8478-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ISCER55570.2022.00015

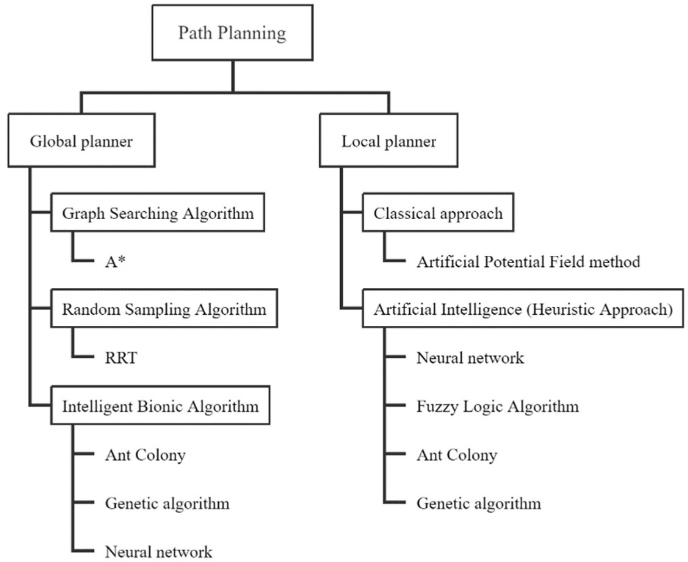


Fig. 1. A categorization of path planning methods discussed in this survey.

According to Chik's research [8], path planning methods can be separated into two categories which are the global planner and local planner. The global planner [9] generates the route by setting a global map, a start point, and an endpoint. Before navigation, the global planner must first process a known or partially known static map of the environment. Then, the navigation methods can be divided into three parts by Cai's research [10] including the graph search-based algorithm, the random sampling algorithm, and the intelligent bionic algorithm. The graph searching algorithm plans the path based on the grids in the map, while the random sampling algorithm plans a path based on random points generated in the map. The intelligent bionic algorithm is an interesting kind of method that gains inspiration from creatures or natural behaviors. For the local planner [9], a mobile robot mainly focuses on its surrounding information and generates a local path accordingly. For comparison, the main difference between the global planner and the local planner is whether the map contains the overall environment information. In terms of the local planner, Zafar [11] further classified it into the classical approach and the artificial intelligence technique. The classical approach is highly dependent on accurate data from the surrounding environment to execute the path planning algorithms, while the artificial intelligence technique can be adaptable to the uncertain surrounding information such as in a dynamic environment. According to the above-mentioned classifications and the typical examples that this paper further explains, the overall classification chart of path planning methods is in Fig. 1.

In this paper, we classify the path planning method into four different types which are classical methods, bionic methods, artificial intelligence methods, and hybrids. Classical methods of path planning rely on known knowledge in the environment to generate a path [12]. We choose A* as a graph searching algorithm, RRT as a random sampling algorithm, and APF as a local planner for the classical methods in chapter II. These three methods are commonly used and were introduced before 2000. Therefore, they are also considered classical methods. The Bionic method uses some characteristics in creatures such as the ant colony method studies the habit of ant searching food and the genetic algorithm uses the pattern of heredity which are presented in chapter III. Artificial intelligence (AI) methods can solve path planning problems in an unknown or partially known environment. The algorithms we choose in the AI method which is explained in chapter IV are fuzzy logic and neural network based on fuzzy mathematics and machine learning respectively. In Chapter V, we introduce an emerging classification called the hybrid method, which was popularly proposed in recent years. This kind of method can mix the advantages of different path planning algorithms to further optimize the performance.

II.CLASSICAL METHODS

Classical methods lay a solid foundation for the solution of path planning problems and give essential insights to future research. In this chapter, three typical methods as A*, RRT, and APF are chosen to represent classical methods and their variants are also further explained. The explanation of these methods is chronologically narrated and summarized with ancillary tables, flowcharts, and other visual aids.

A.A and its variants*

A* [13] [14] is a kind of graph searching algorithm to search the shortest route which was introduced in 1968 by Hart and his fellows [1], while it needs high memory and time cost. In the period of path planning, A* utilizes the cost function:

$$f(n) = g(n) + h(n), \tag{1}$$

to evaluate the quality of the node. The cost function in Eqn.1 shows the estimated distance cost from the current node to the goal node. A* algorithm chooses the minimum value of the cost function as the next expended point. In Eqn.1, *n* is the closest node during the path planning period. *g(n)* is the shortest path from a start node to *n* node. *h(n)* is the prediction value of the heuristic function such as Manhattan distance, Euclidean distance, and diagonal distance from the *n* node to the goal point. A* and its variants are global planners which require a known or partly known map before planning the path, while the variants of A* improve the performance of A* or add a new function to A*. Anytime A*(ARA*) [15] [16] can change its performance boundary based on the available searching time. D* algorithm [17] [18] can change the path dynamically by the previous calculation path solution. Theta* [19] [20] improves the distance consequence of A* while Theta* may lead to more time consumption. The specific table of pros and cons of different algorithms is in Table I.

TABLE I. ADVANTAGE AND DISADVANTAGE OF VARIANTS OF A*

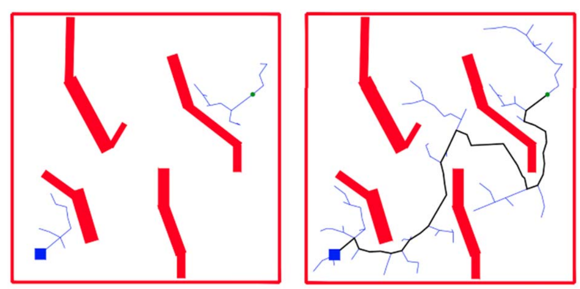
Algorithm	Advantages	Disadvantages
A*	Apply graph theory on a discretized configuration space with complete resolution.	High memory and time cost.
ARA*	An efficient anytime heuristic search.	Ineffectively cost the effort in deeper areas of the search space.
D*	It can plan the path in the changing, known, unknown areas with an efficient and optimal method.	The method can only work on the point which is restricted to the change of cost.
Theta*	The path is a shorter distance.	It costs more time.

In 2020, Qi and his fellows present an algorithm called improved A* [13] which has shorter computing time, fewer expansion nodes, and more targeted pathfinding than A*. Compared with A*, improved A* optimize four aspects: (1) improved A* considers the goal direction's quadrant without calculating every direction, (2) improved A* can detect whether the obstacle exists on the line between expanded node and goal when the expanded node is close to the goal point. If there is no obstacle on the line, improved A* can exit the procedure of A* which can decrease the cost of time, (3) improved A* changes the heuristic function to the combination of the node to its last generation parent node, and (4) improved A* uses the simulated annealing method to optimize the selection of the expanded nodes.

Compared to A*, the distance of generated path decreases by 3%, time consumption reduces by 88% and the number of grids experienced in the path planning process decreases by 92.5% in the research of improved A*. So, the improved A* can use less time to plan the path better than A*.

B.RRT and its variants

LaValle introduced RRT in 1998 [2] [21] [22] [23] which is a random sampling algorithm and global planner that can generate the path rapidly and flexuously. The strategy of the RRT algorithm is to build a tree from start point to goal point based on random sampling in the surrounding environment. The tree is expanded by increasing the number of new vertexes which is close to the created random vertex. RRT can plan the path with a high coverage rate and wide search area so that RRT can create the path in the unknown environment to the greatest extent feasibility. RRT-Connect was developed in 2000 [24] [25] which has two rapidly-exploring random trees rooted in the start and goal point which can improve the search precision. The schematic of the RRT-Connect is in Fig. 2. In Fig.2, two trees (blue lines) grow in the start point (blue square) and the goal point (green circle) in Fig. A. Then, two trees are joined to generate the path (black line) that can avoid obstacles (red shape) as shown in Fig. B. In 2011, Karaman and Frazzoli designed the RRT* [21] [26] to improve the performance in the high dimension of RRT and the path quality, introduce new important elements of close neighbor search and rewiring procedures. However, this method has some limitations such as slow convergence rate, Jagged and suboptimal paths, and large memory consumption. In 2012, RRT*-Smart [21] [27] was introduced to improve the efficiency of time and cost by two new features called intelligent sampling and path optimization. The limitation of the RRT*-Smart is that it needs careful adjustment of the intelligent sampling frequency in different environments. Similarly, Reza Mashayekhi and his fellows introduced another improved RRT* algorithm called Informed-RRT* [21] [28] [29] in 2020. This new algorithm can accelerate the convergence based on the ellipsoidal informed subset while it can only shrink planning problems in specific conditions because of the initial solution cost. In 2021, Wang introduced KB-RRT* algorithm [14] [30] to plan the path in a differential drive mobile robot with fast and high quality. The next step of KB-RRT* is to improve the performance by using the learning method, refine the generated path to be utilized in autonomous driving, and combine the KB-RRT* with inverse reinforcement learning methods to meet social requirements in the environment in which humans and robots both exist. The advantages and disadvantages of the variant of RRT are in Table II.



A B

Fig. 2. A RRT-Connect applied path planning [24].

TABLE II. ADVANTAGE AND DISADVANTAGE OF VARIANTS OF RRT

Algorithm	Advantages	Disadvantages
RRT	Rapid path generation	Flexuous path
RRT-Connect	Improved search precision	Deficient in smoothness and path length
RRT*	Guarantee probability of pathfinding to 1	Expensive in high dimensions

Informed-RRT*	Accelerate convergence speed	Requiring extra time to attain goal configuration.
RRT*-Smart	A faster rate of optimum solution and reduced execution time.	The frequency of intelligent sampling must be carefully adjusted in a different environment.
KB-RRT*	Fast and high-quality path planning with better performance	Low search accuracy.

In 2021, a new improved RRT* called Fast-RRT* (F-RRT*) was developed. As illustrated by Bin Liao and his fellows [31], RRT* gives asymptotic optimality but is limited due to its sluggish convergence rate and costly starting solution. Based on all variables considered in Table III. The quality of the initial solution can be improved by raising the radius parameter R_{near} because a high radius R_{near} makes it easier to minimize the Cost (X_{rand}). However, it rapidly increases the number of neighboring vertices and computing time. As a result, when F-RRT* adds X_{rand} to the tree, the radius parameter R_{near} is removed. F-RRT*creates a parent node for the random point instead of selecting it among the existing vertices, resulting in a superior initial solution that converges faster and reduces path cost. The creation process can be broken down into two steps: (1) the first one is to connect the X_{rand} and the parent node of $X_{reachest}$ simultaneously without collision by using the 'find the next point' algorithm as illustrated in. The algorithm shows the logic of how to avoid collision and connect the X_{rand} and $X_{reachest}$ by loop and judgment. (2) The second one is to look for the ancestors of $X_{reachest}$ and search for the vertex $X_{reachest}$ which can be connected to X_{rand} without collision. The algorithm procedures of the second step are: (1) create new parameter $D_{dichotomy}$, (2) use dichotomy process to create X_{create} based on X_{rand} and $X_{reachest}$, (3) judge upon termination condition based on $D_{dichotomy}$ and (4) process until the desired X_{create}.

Both of which take little calculations and the triangle inequality is utilized repeatedly throughout the process, resulting in pathways that outperform RRT*. The simulation result compared with RRT* [26], Q-RRT* [32], RRT*-smart [27] is shown in Fig. 3.

TABLE III. VARIABLES AND FUNCTIONS CONSIDERED IN F-RRT* ALGORITHMS

Variables / Functions	Explanation
R_{near}	Radius parameter
X	The configuration space
X_{obs}	The obstacle region
X_{free}	X_{free} = X/X_{obs} is the obstacle-free region
X_{rand}	In each iteration, a sample X_{rand} was chosen randomly from X_{free}
E	The set of connection relationships between the vertexes
V	A set of vertexes
G	RRT explores the space through a tree G = (V, E)
$X_{nearest}$	The closest vertex to X_{rand}
$X_{reachest}$	The vertex connected to X_{rand}
X_{start}	The initial state of a path
Cost()	Given a vertex xp from V. it returns the full length of the path from X_{start} to xp.
Parent()	Given a vertex xp from V, it returns the parent vertex of xp

ALGORITHM 1: FIND THE NEXT POINT

```
Input    G    x_nearest    x_rand
Output   x_nearest
x_reachest ← x_nearest
while    x_reachest ≠ x_nearest  do
  |   if   CollisionFree(x_rand, Parent(x_reachest))   then
  |     |   x_reachest ← x_nearest
  |   else
  |     |   Return x_reachest
  |   end
end
Return x_reachest
```

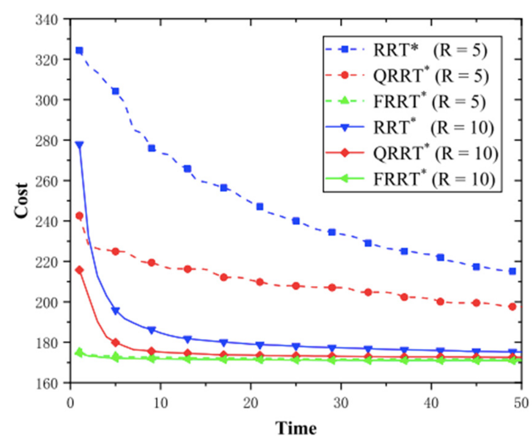


Fig. 3. A simulation result of F-RRT* applied path planning [31].

C. Artificial potential field and its variants

In 1986, Khatib proposed the artificial potential field approach as a virtual force method [33] which is a kind of local planner. The main idea is to use an artificial potential field to simulate the influence of a target and an obstacle on robot mobility. The target's potential energy is low, whereas the obstacles are high. The attraction of the target to the robot and the repulsion of the obstacle to the robot are generated by the potential energy difference, and the combined force guides the robot to move along the negative gradient of the potential field until it reaches the target point. The basic principle of this method is shown in Eqn.2:

$$U_q = U_{att}(q) + U_{rep}(q), \quad (2)$$

where U_{att} is the gravitational field, which guides the robot to the target position; U_{rep} is a repulsive force field that guides the robot to avoid obstacles. The artificial potential field method makes calculations easier and allows for the creation of a safe and smooth path. The complicated potential field environment, on the other hand, may yield local minima outside the target point, making the robot trapped in oscillatory motion and unable to reach the target because sometimes attractive fields and repulsive fields may offset. To address these problems, researchers have improved the potential field method in the following decades. In 1992, Rimon and Koditschek's navigation function [34] can eliminate local minima because it generates a bounded-torque feedback controller for the robot's actuators that assures collision-free motion and convergence to the destination from practically all initial unconstrained configurations. In 2000, Yunfeng Wang and G. S. Chirikjian proposed a new potential field method [35] for robot path planning that introduced variable K which used a simple rectangular grid to discretize a workspace, both simplifying computational domain and facilitating grid generation. In 2014, L. Zhou and W. Li developed an enhanced obstacle potential field function model [36] that takes into account the size of the robots and obstacles, and adaptively alters the weight of the obstacle potential field function to help the robot escape local minima. In 2019, to further solve the local minimum, the activation of a virtual escape force when a local minimum is observed is offered as a unique repulsive potential function by Ameni Azzabi and Khaled Nouri [37]. This force acts as a rotating force, allowing the robot to break free from stalemate positions and turn smoothly away from obstacles toward the destination.

In recent years, several emerging path planning approaches based on artificial potential have been proposed. In 2020, Elia Nadira Sabudin and his fellows [38] proposed an improved APF and a path pruning technique to shorten the path accordingly and in their case, the robot removes the waypoints to avoid repetition, and finds the lowest point from the midpoint, both overcoming the local minima problem and successfully reduces the path length.

III.BIONIC METHODS

Traditional path planning algorithms have some defects in a complex environment. For example, the artificial potential field method may fall into a local minimum point, and the A * algorithm has a large amount of computation. Scholars propose bionic algorithms to improve traditional path planning algorithms. Bionic algorithms are intelligent computing method that imitates the intelligent behavior of biological groups or the structure and function of organisms. In general, bionic algorithms find a globally optimal solution in the complex solution space, which has the characteristics of random, parallel, and distributed [39].

A.Ant colony and its variants

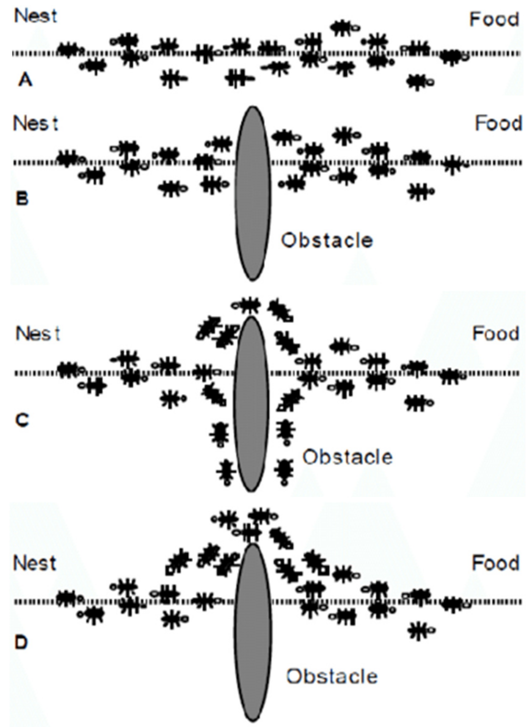


Fig. 4. An ant colony obstacle avoidance behavior applied path planning.

Ant colony algorithm (ACO) is a kind of bionic algorithm, which is an algorithm that imitates the process of an ant colony looking for food. It was proposed by Italian scholar Dorigo M et al. in 1991. AG was first used to solve the traveling salesman problem (TSP), and it is better in discrete optimization problems [40].

The behavior of the ant colony is shown in Fig. 4. When the ants walk, they leave pheromone on the ground, which is used to mark the walking path. Pheromone is a substance that volatilizes over time and can guide ants to a path with a higher concentration. In the absence of pheromone, the possibility of ants choosing the upper and lower roads in part C of Fig. 4 is the same, but ants walking on the shorter path above can walk more back and forth at the same time, leaving a stronger pheromone. The stronger pheromone eventually guides the ant colony to choose a shorter upper path [41]. And the ant colony behavior is also applicable to the situation where the obstacles dynamically appear on the path of the ant colony as shown in part B of Fig. 4, which means that the real-time ant colony path planning algorithm is also possible [42].

The ant colony algorithm summarized by the behavior of the ant colony is a heuristic algorithm iteratively, which can be roughly regarded as the following process. In each iteration, multiple ants are set up and searched separately in parallel. After each ant completes a search, it releases pheromone on the way, and the amount of pheromone is proportional to the quality of the solution. The path of the ant is selected according to the intensity of the pheromone (the initial pheromone amount is set to be equal) while considering the distance between two points, a random local search strategy is adopted. This makes the amount of pheromone on the shorter path larger, and the probability of later ants choosing this path is also greater. Each ant can only take a legal route (cannot cross obstacles, etc.), for which a taboo table is set to control. All ants search once and iterate once. Every iteration updates all the pheromones. The updated pheromone includes the evaporation of the original pheromone and the increase of the pheromone along the path. When the predetermined number of iterative steps is reached, or there is a stagnation phenomenon (all ants choose the same path, the solution does not change), the algorithm ends, and the current optimal solution is used as the optimal solution of the problem [43]. The flowchart of a typical ant colony path planning algorithm is shown in Fig. 5, showing more detailed algorithm operation steps.

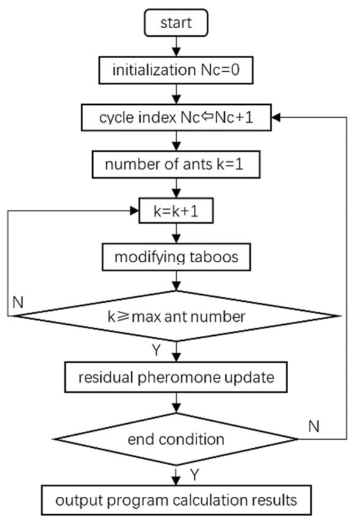


Fig. 5. A flowchart of the ant colony algorithm applied path planning.

B.Genetic algorithm and its variants

Genetic algorithms are search algorithms based on Darwin's theory of evolution's natural selection principles to find the survival of the fittest. In 1868, Charles Darwin proposed a method for the transmission of hereditary information across cells and generations in his Pangenesis theory [44]. In principle, natural selection and genetic processes such as recombination and mutation are used to create chromosomes that are more fit. Variables are expressed on a chromosome as genes. On the response surface of genetic algorithms, there is a population of potential solutions. In 2003, Jianping Tu and his fellows [45] introduced a novel coding method with variable-length binary strings to substitute the traditional representation of the genotype (with a string of bits). The flowchart of their method is in Fig. 6. However, the random mutation operators in conventional GA methods may cause infeasible paths. So in 2012, instead of the direction of motion via the mutated node, AdemTuncer and his fellows [46] took the node based on the fitness value of the complete path and thus derived an improved GA method that converged more rapidly than the previous methods did. In 2018, B.K.Patle and his fellows [47] introduced a new variant to the GA method by using the binary codes through the matrix. The simulation results showed this method presented a better way of sequencing, ordering, selecting, and grouping variables. In 2020, The multi-population migration genetic algorithm (MPMGA) proposed by Hao Kun and his fellows [48] split a big population into multiple tiny populations with the same population number at random. The MPMGA is not only applicable for simulation maps of varied sizes and barrier distributions, but also breaks the local optimal solution, solves the significant homogeneity of population individuals, increases the quality of convergent individuals, and speeds up the algorithm's convergence time.

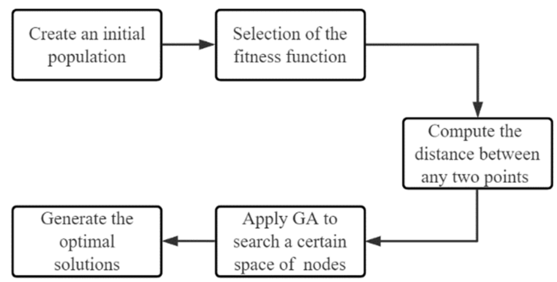


Fig. 6. A flowchart of GA algorithms applied path planning.

IV.ARTIFICIAL INTELLIGENCE METHODS

The core of artificial intelligence path planning algorithm is to imitate people's thinking mode, which usually has high learning ability. Learning ability means that the algorithm is evolutionary and could achieve a better operation effect after a lot of training.

A.Fuzzy logic and its characteristic

The algorithm uses the logic of fuzzy classification of problems which like human beings, rather than accurate classification through a threshold, is called fuzzy logic. The path planning algorithm based on fuzzy logic can integrate the return information of heterogeneous sensors and task objectives, to take the form of human-like decision-making for path planning. C.C. Lee explained the principle of fuzzy logic in detail and introduced the controller design using fuzzy logic [49]. Similar to other classical fuzzy logic controllers, the path planning algorithm based on fuzzy logic formulates a path planning rule by putting human experience in path planning and vehicle driving into the fuzzy logic table. In application, the human-like decision-making form can be realized just by making the program make decisions according to the fuzzy table.

A basic fuzzy logic path planning controller is shown in Fig. 7. Firstly, the input quantity, such as the distance measured by the laser rangefinder, is to be blurred. In the process of fuzzification, it is necessary to use the implication function to convert the input value of logic into implication, so as to fuzzy judge which interval the data is in. For instance, if the distance returned by the sensor is 10 millimeters, the implication of “short distance” is 1, “middle distance” is 0, “long distance” is 0. When the distance returned is 30 millimeters, the implication of “short distance” is 0, “middle distance” is 0.5, “long distance” is 0.5. Usually, the mathematical graphics of the implication function is arbitrary, but the common function graphics are triangle or trapezoid, as shown on the left in Fig. 7.

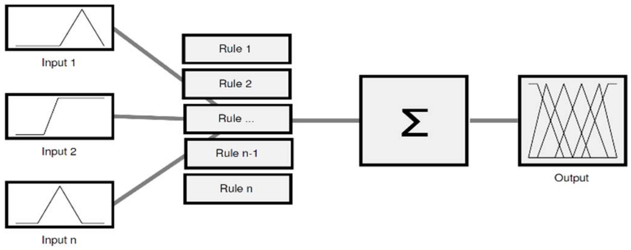


Fig. 7. The structure of a fuzzy logic system applied path planning [64].

Next, several outputs (Fire Strength, FS) are obtained from the fuzzed input through certain rules and operations. Table IV is a typical fuzzy logic path planning rule table written by human experience, which records multiple calculation rules between inputs and outputs.

Finally, de-fuzzification is performed to merge multiple FS into a unique output. The most commonly used method of de-fuzzification is the weighted average decision method. As shown in the following formula：

$$Output = \frac{\sum_i FS_i * OW_i}{\sum_i FS_i}. \quad (3)$$

TABLE IV. FUZZY LOGIC RULE [64]

Neighbours	**Distance**	**Chance**
Low	Far	Very Weak
Low	Medium	Weak
Low	Near	Little Weak
Medium	Far	Weak
Medium	Medium	Little Weak
Medium	Near	Little Strong
High	Far	Little Weak
High	Medium	Little Strong
High	Near	Strong

FS_i is the Fire Strength output in the previous step, and OW_i is the Output Wight. The final decision of the algorithm is the judgment by setting a threshold for $Output$.

Due to the use of fuzzy logic, the scale of the algorithm can be reduced, so as to design an efficient and low computational path planning algorithm. Therefore, the path planning algorithm based on fuzzy logic is considered to be more suitable for real-time dynamic environment path planning [50]. And a well-designed fuzzy logic table can enable the algorithm to overcome the problems such as local poles which are easily produced by the potential field method and make the algorithm achieve high robustness. Therefore, fuzzy logic path planning can be better used in an unknown environment [51]. At the same time, due to the limitations of human experience, experience-based algorithms always have some shortcomings. When the input quantity becomes more, to be more specific when there are more obstacles in the map, the reasoning rules, and fuzzy tables expand sharply, resulting in a significant increase in the demand for computing power. Fuzzy logic path planning also has the phenomenon of symmetric indecision, which may cause the robot to deadlock when facing symmetrical objects [52].

B.Neural network and its variants

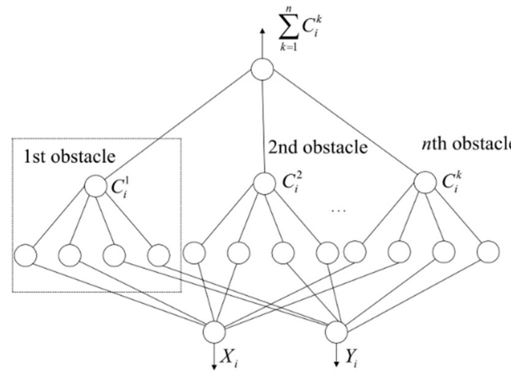


Fig. 8. A neural network applied path planning [65].

The neural network is an algorithm that appeared in the 1940s, which is the basic algorithm for deep learning. The neural network algorithm can fit the mapping relationship between multi-dimensional input and output through a pre-designed multi-layer neural network. The process of putting some certain input and output into the neural network and continuously approximating the mapping relationship is called training. The main purpose of intensive training is to make the weight of each branch in the neural network more accurate. The enhancement of the accuracy of neural network algorithms through training is called learning, which is the biggest feature of neural network algorithms. Path planning is also a mapping from perception space to behavior space. The mapping relationship can be expressed in different ways, but it is difficult to express it with precise mathematical equations. The neural network cleverly avoids this problem and achieves the purpose of establishing an accurate model through the self-learning of the network.

Roy Glasius, Andrzej Komoda, and Stan C. A. M. Gielen detailed the use of neural networks for path planning and obstacle avoidance in 1995 [53]. Fig. 8 is a typical neural network used in the establishment of a two-dimensional environment model in a neural network path planning algorithm. The modeling of this environment includes n obstacles, x_i and y_i represent the coordinates in the two-dimensional environment. When the output c_i^k (0 or 1) is 1, the robot collides with an obstacle. And when c_i^k is 0, there is no collision. So, when $\sum_{k=1}^{n} c_i^k$ is 0, it represents a collision-free path.

In 2000, Simon X. Yang and Max Meng proposed a dynamic path planning algorithm based on a biologically inspired neural network, which considers safety factors so that the planned path is not too close or too far [54]. In 2004, the study of Simon X. Yang and Chaomin Luo implemented Complete coverage path planning (CCPP) using neural networks [55]. CCPP is a big branch in the field of path planning, which is often used in cleaning or painting robots. In 2009, Hong Qu, Simon X. Yang, Allan R. Willms, and Zhang Yi presents a modified pulse coupled neural network (PCNN) model for real-time path planning. In theory, the path generated by the algorithm is always the globally shortest path from the robot to the target [56]. In 2017, Rehder E, Quehl J, and Stiller C use a convolutional neural network (CNN) to imitate human path planning mode, which is used to realize the path planning of complex traffic conditions in the real world [57]. In 2020, Keyu Wu, Han Wang, Mahdi Abolfazli Esfahani, and Shenghai Yuan apply deep neural network (DNN) to path planning. The calculation time and effectiveness of the algorithm are independent of environmental conditions, showing its superiority in large-scale and complex environments [58]. In 2021, Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok apply the graph neural network (GNN) to the path planning of multi-robots, so that the CNN extracted from the local observation can be transmitted between the multi-robots [59]. So that robot can navigate in the restricted workspace.

V.HYBRID METHODS

In recent years, researchers have mixed different features of several path planning algorithms to generate the hybrid method with optimal performance. The characteristics of hybrid methods are a combination of advantages from classical or intelligent path planning algorithms. This chapter focuses on A* in the hybrid methods and other hybrid methods are also explained to showcase the development in recent years because A* has a fast search speed and it is common in hybrid methods.

*A.Hybrid algorithms based on A**

Because the heuristic method is in the A* algorithm, which can be combined with different algorithms such as RRT [60], ant colony algorithm [61] , and artificial potential field method [62] to plan the shortest path.

1)Optimized RRT-A:* In 2019，Ayawli and his fellows introduced an optimized rapidly exploring random tree A* (ORRT-A*) algorithm *[60]* based on morphological dilation (MD), goal-biased RRT, A*, and cubic spline algorithms to improve the performance of RRT-A* including less time consumption and less experienced node number. In this research, the speed of the generation tree based on goal is higher by the extra step size. A* algorithm is used to obtain the shortest route by choosing the nearest node in the tree generation period. Before generating the path, the MD technique and cubic spline interpolation are used to smooth path planning and avoid collisions between obstacles and a robot. The whole algorithm has five steps which are obtaining and processing map, computing the RRT roadmap, path query, and optimization, path smoothening, and path replanning. In step 1, the map is generated by a sensor that can provide a workplace to run the algorithm. It is worth mentioning that the MD technique is used to inflate some obstacles for safety issues. In step 2, the roadmap is produced by the modified goal-biased RRT. In step 3, the route is planned by the RRT roadmap's initial point to the target at first. Then, the A* algorithm optimizes the path by selecting the nearest node during the process of the tree. In step 4, the cubic spline interpolation enables the route to become smoother after the A* optimization. In step 5, when a random obstacle occurs during the robot navigation period, the robot replans the route by detecting the angle and position of the obstacle. The overview flow chart of this algorithm is in Fig. 9. According to the result, it shows the benefit of a safe and shortest path including narrow passages, sparse and dense environment. In the future, the research team applies this method to the 3D field *[60]*.

2)Improved A-star Ant Colony Algorithm: In 2021, Lan and his fellow introduced an improved A-star ant colony algorithm [61] which can tackle the problem of slow convergence speed, high time consumption, and large number of iterations in the ant colony algorithm. With the usage of A* and its directionality, this algorithm optimizes the initial pheromone and pheromone enhancement coefficient of the ant colony to improve the convergence and avoid the algorithm falling into the local optimum. The improvement of the pheromone giving mechanism is changed to enhance global searching ability. Apart from that, this algorithm can work in a complex environment by the combination of grid environment information and the set of environmental penalty coefficients. Compares to the traditional ant colony algorithm results, this method decreases 14% in the path distance with higher searching time and convergence speed which can work better in a complex environment. In Fig. 10, the black lines in the two figures are the generated path and the black blocks are obstacles. The comparison result shows that the generated path of the improved A-star ant colony algorithm is smoother than the path in the traditional ant colony algorithm.

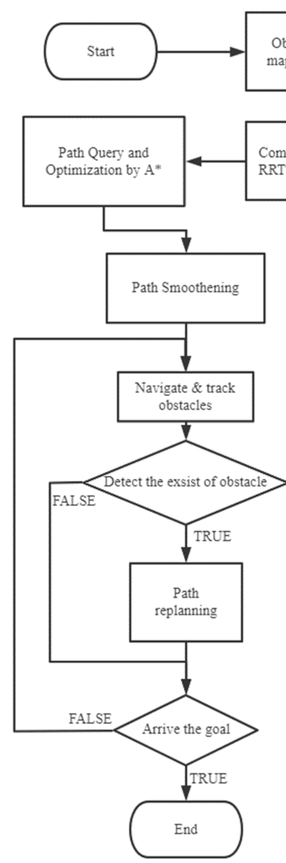


Fig. 9. A flowchart of the main step in the RRT-A* applied path planning.

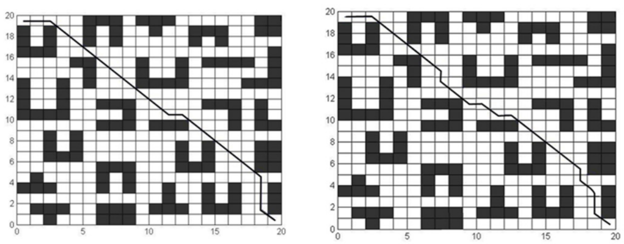


Fig. 10. A comparison applied path planning of the improved A-star ant colony algorithm (figure A) and the traditional ant colony algorithm (figure B).

3)Improved heuristic function of A-Star combined with artificial potential field method: In 2021, Zhang and his fellows proposed a new method *[62]* that combined A* with the artificial potential field. Based on artificial potential fields, improved distance heuristics, and B-spline path smoothing, Zhang's method takes surrounding obstacles into consideration and gets a smoother path. The valuation function is changed which is presented as follows:

$$f(n) = g(n) + h'^{(n)} + v(n). \tag{4}$$

In Eqn. 3, $f(n)$ is the total estimated cost. $g(n)$ is the movement cost from the start point to the current point. $h'(n)$ is distance cost which is changed to Diagonal distance from Manhattan distance which is used in traditional A*. Compared with Manhattan distance and Euclidean distance, Diagonal distance can plan the shortest path in moderate time consumption as the distance heuristic cost after Zhang's verification. $v(n)$ is the potential field cost which includes the obstacles information around the current point. According to the artificial potential field, Zhang's method calculates the resultant of obstacles' repulsion force and target's attraction force first. Then, calculate the projection value on eight adjacent expansion nodes as $v(n)$ because A* algorithm uses eight directions on grids to search path. Because Zhang's method includes obstacles and distance information, it has a reduction of turning points' quantity and improvement of the execution efficiency and smoother path compared with A*.

B.Other Hybrid algorithms

Apart from the hybrid algorithm based on A*, there are some hybrid methods is the combination of other algorithms such as the combination of APF and genetic algorithm and the mixture of fuzzy control and ant colony algorithm.

1)Membrane evolutionary artificial potential field: In 2019, Rosas and his fellows introduced a membrane evolutionary artificial potential field (memEAPF) method *[63]* for robot path planning. memEAPF is a kind of hybrid algorithm with the combination of membrane algorithm, genetic algorithm, and artificial potential field method. In the first step of the memEAPF algorithm, each individual is evolved by a basic membrane made up of an evolutionary artificial potential field (EAPF) which is. The EAPF is a kind of genetic algorithm utilizing APF as a fitness function to generate parameters that can discover the best individual in each basic membrane. In the second step, all of the basic membranes converge into a single membrane that contains all of the individuals based on communication rules which can find the best individuals. In the third step, the process is performed several times to fine-tune the sets of parameters, aiming to obtain optimal or near-optimal path planning. The three steps are illustrated in Fig. 11. After experiments, memEAPF shows the ability to find the shortest path with low time consumption in the static or dynamic environment compared with other methods including parallel evolutionary artificial potential field, pseudo-bacterial potential field, and bacterial potential field. Apart from that, memEAPF is a high-performance highly scalable method that computers can use in sequential mode to plan the path in a complex environment.

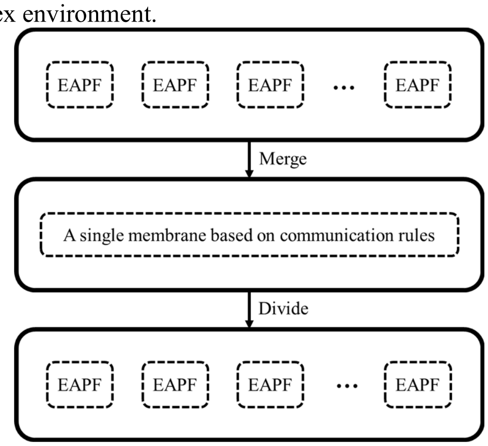


Fig. 11. A membrane structure of the memEAPF applied path planning.

2)Fuzzy gain-based dynamic ant colony optimization: A fuzzy gain-based dynamic ant colony optimization (FGDACO) *[64]* was proposed by Sangeetha and his fellows in 2021. This algorithm focuses on planning a safe and smooth path in a dynamic environment based on a combination method of ant colony algorithm and fuzzy control. According to the ant colony algorithm, the pheromone trail and the heuristic are two vital parts relating to exploring and exploiting the path. The pheromone trail is improved by utilizing a sigmoid function-based with relative distance as a gain function to prevent unexpected traversals. Fuzzy control is based on approximate reasoning and Mamdani fuzzy inference system to avoid collision and provide a route. Based on Sangeetha's test, FGDACO has a shorter path distance, faster convergency and less deviation in its independent tests compared with a cuckoo optimization algorithm, fuzzy-genetic algorithm, and fuzzy logic based ant colony optimization. FGDACO can be used in dynamic real-time road networks and vehicle routing changing problems. In Fig. 12, the behavior is presented by the sequence of the figures A, B, and C. The orange rectangles and the black arrows show the moving obstacles and their moving direction, while the black rectangles are the static obstacles. The blue line is the generated path by FGDACO.

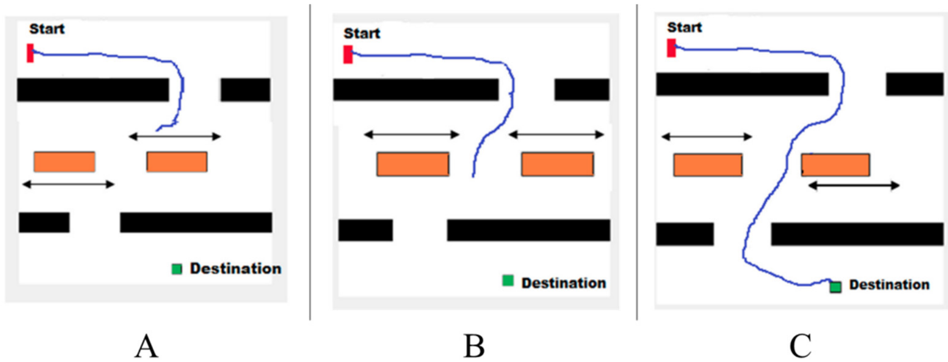


Fig. 12. A collision avoidance behavior of the FGDACO applied path planning in the dynamic environment [64].

VI.CONCLUSION

In this paper, we summarize mobile robot 2D path planning algorithms into three stages: classical algorithm stage, imitating algorithm stage, and hybrid algorithm stage. The earliest classical algorithms, such as A* and RRT, are usually based on a certain mathematical model. Classical path planning algorithms can find the shortest collision-free path in specific cases, but due to many preconditions required by the algorithm, classical algorithms are usually lacking flexibility. After that, scholars began to imitate some mature systems for path planning, such as the earlier artificial potential field method imitating physical laws, the later ant colony algorithm imitating biological group action, and the neural network algorithm imitating human thinking form in recent years. These imitation algorithms have high flexibility and even self-learning habits, but they usually cannot solve the problems of a large amount of calculation, are easy to fall into local optimal solution, or are unable to converge stably at the same time. In order to integrate the advantages and discard the disadvantages of the above algorithms, hybrid algorithms are the most popular in recent years. At present, unless the computing power is greatly improved, the hybrid algorithm is a feasible direction of path planning.

REFERENCES

[1] N. J. N. A. B. R. PETER E. HART, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE TRANSACTIONS OF SYSTEMS SCIENCE AND CYBERNETICS, pp. 100-107, 1968.

[2] S. M. LaValle, " Rapidly-exploring random trees: A new tool for path planning," Report No. TR 98-11, 1998.

[3] Y. K. a. N. A. Hwang, "A potential field approach to path planning," IEEE Transactions on Robotics and Automation, pp. 23-32, 1 8 1992.

[4] C.-L. C. a. C.-C. C. Ying-Tung Hsiao, "Ant Colony Optimization for Best Path Planning," lntematbnalSymposium on Communications and Information Technobgies, pp. 109-113, 2004.

[5] J. T. a. S. X. Yang, "Genetic Algorithm Based Path Planning for a Mobile Robot," lEEE lnlernatioarl Conference on Robotics & Automation, pp. 1221-1226, 2003.

[6] J. Y. a. N. Pfluger, "Path planning and execution using fuzzy logic," Navigation and Control Conference, pp. 1691-1698, 8 1991.

[7] A. K. A. S. C. A. M. G. RoY GLASIUS, "Neural Network Dynamics for Path Planning and Obstacle Avoidance," Neural Networks, pp. 125-133, 1995.

[8] C. F. Y. E. L. M. S. T. Y. L. Y. S. P. J. H. C. S. F. Chik, "A Review of Social-Aware Navigation Frameworks for Service Robot in Dynamic Human Environments," Journal of Telecommunication, Electronic and Computer Engineering, pp. 41-50, 2016.

[9] H. M. Zhuozhen Tang, "An overview of path planning algorithms," Earth and Environmental Science, 2021.

[10] C. W. J. C. S. S. Kuanqi Cai, "Mobile Robot Path Planning in Dynamic Environments: A Survey," arXiv preprint, p. 2006.14195, 25 6 2020.

[11] J. C. M. Mohd. Nayab Zafar, "Methodology for Path Planning and Optimization of Mobile Robots: A Review," Procedia Computer Science, pp. 141-152, 2018.

[12] S.-M. A. Mohd Nadhir AbWahab, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?," Annual Reviews in Control, pp. 233-252, 2020.

[13] Q. X. J. Jian'an, "Path planning for unmanned vehicle based on improved A* algorithm," Journal of Computer Applications, pp. 2021-2027, 10 7 2020.

[14] B. L. M. Q.-H. M. Jiankun Wang, "Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning," Expert Systems with Applications, vol. 170, no. ISSN 0957-4174, 2021.

[15] G. G. a. S. T. Maxim Likhachev, "ARA*: Anytime A* with Provable Bounds on Sub-Optimality," Advances in neural information processing systems, pp. 767-774, 2003.

[16] J. T. T. a. W. R. Silvia Richter, "The Joy of Forgetting: Faster Anytime Search via Restarting," In Twentieth International Conference on Automated Planning and Scheduling, 4 2010.

[17] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1993.

[18] R. K. H. A. H. Masoud Nosrati, "Investigation of the * (Star) Search Algorithms: Characteristics, Methods and Approaches," World Applied Programming, pp. 251-256, 2012.

[19] A. N. a. K. D. a. S. Koenig, "Theta*: Any-Angle Path Planning on Grids," AAAI, 2007.

[20] M. S. S. a. H. K. Zeyad Abd Algfoor, "A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games," International Journal of Computer Games Technology, 2015.

[21] A. K. Z. H. Iram Noreen, "Optimal Path Planning using RRT* based Approaches: A Survey and Future Directions," International Journal of Advanced Computer Science and Applications, pp. 97-107, 2016.

[22] B. M. Q.-H. JiankunWang, "Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning," Expert Systems With Applications, 25 12 2020.

[23] R. B. D. D. A. A. N. B. a. P. F. Omar SOUISSI, "Path Planning: A 2013 Survey," Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM), pp. 1-8, 2013.

[24] J. ,. S. M. L. James J. Kuffner, "RRT-Connect : An Efficient Approach to Single-Query Path Planning," International Conference on Robotics & Automation, pp. 995-1001, 4 2000.

[25] H. W. C. a. H. F. Jin, "Improved RRT-Connect Algorithm for Urban low-altitude UAV Route Planning," Journal of Physics: Conference Series, vol. Vol. 1948., no. No. 1. IOP Publishing, 2000.

[26] E. F. S. Karaman, "Sampling-based algorithms for optimal motion planning," The International Journal of Robotics Research, Vols. pp. 846-894, no. 30 (2011), 2011.

[27] J. N. U. M. Y. A. a. O. H. Fahad Islam, "RRT*-Smart: Rapid convergence implementation of RRT* towards optimal solution," International Conference on Mechatronics and Automation, pp. 1651-1656, 2012.

[28] S. S. T. B. J.D. Gammell, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in 2014 IEEE/RSJ international conference on intelligent robots and systems, 2014.

[29] M. Y. I. I. M. H. A. I. A. A. I. A. REZA MASHAYEKHI, "Informed RRT*-Connect: An Asymptotically Optimal Single-Query Path Planning Method," IEEE Access, pp. 19842-19852, 31 1 2020.

[30] Y. G. Y. T. H. C. T. F. a. Y. H. Longfei Jia, "A Novel Path Planning Method Based on Regional Index Algorithm for Hyper-redundant Manipulator," International Conference on Intelligent Robotics and Applications, pp. 669-679, 20 10 2021.

[31] B. W. F. H. Y. M. R. Z. S. &. Q. Liao, "F-RRT*: An improved path planning algorithm with improved initial solution and convergence rate ☆," Expert Systems With Applications , no. 0957-4174, 2021.

[32] I.-B. a. L. S.-J. a. K. J.-H. Jeong, "RRT*-Quick: A Motion Planning Algorithm with Faster Convergence Rate," in Robot Intelligence Technology and Applications 3, Cham, Springer International Publishing, 2015, pp. 67--76.

[33] K. O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots[C]," IEEE International Conference on Robotics and Automation, pp. 90-98, 2003.

[34] D. K. E. Rimon, "Exact robot navigation using artificial potential functions," IEEE Trans. Robotics Autom., vol. 8, pp. 501-518, 1992.

[35] Y. W. a. G. S. Chirikjian, "A new potential field method for robot path planning," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation, vol. 2, no. doi: 10.1109/ROBOT.2000.844727, pp. 977-982, 2000.

[36] L. Z. a. W. Li, "Adaptive Artificial Potential Field Approach for Obstacle Avoidance Path Planning," in 2014 Seventh International Symposium on Computational Intelligence and Design, 2014, pp. 429-432.

[37] A. A. a. K. Nouri, "An advanced potential field method proposed for mobile robot path planning," Transactions of the Institute of Measurement and Control, vol. 41 (11), no. 10.1177/0142331218824393, pp. 3132-3144, 2019.

[38] S. E. e. al., Improved Potential Field Method for Robot Path Planning with Path Pruning., vol. 666, Singapore: Springer, Singapore, 2021.

[39] P. V., Path Planning for Vehicles Operating in Uncertain 2D Environments, Butterworth-Heinemann, 2017.

[40] B. M. S. T. Dorigo M, "Ant colony optimization," IEEE computational intelligence magazine, pp. 28-39, 1(4) 2006.

[41] K.-L. S. J.-H. G. C. C. Song-Hiang Chia, "Ant Colony System Based Mobile Robot," in Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference, Dec. 2010.

[42] Z.-q. Z.-x. C. Wen, "Global path planning approach based on ant colony optimization alorithm," JOURNAL-CENTRAL SOUTH UNIVERSITY OF TECHNOLOGY-ENGLISH EDITION- 13.6, p. 707, 2006.

[43] V. R. U. Selvi, "Comparative analysis of ant colony and particle swarm optimization techniques," International Journal of Computer Applications 5.4, pp. 1-6, 2010.

[44] Y. Zou, "Charles Darwin's Theory of Pangenesis," Embryo Project Encyclopedia, 2014.

[45] &. Y. S. X. Jianping Tu, "Genetic algorithm based path planning for a mobile robot," in IEEE International Conference on Robotics and Automation , Taipei, 2003.

[46] A. T. a. M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," Computers & Electrical Engineering, vol. 38, no. 8, pp. 1564-1572, 2012.

[47] B. P. a. D. P. a. A. J. a. S. K. Kashyap, "Matrix-Binary Codes based Genetic Algorithm for path planning of mobile robot," Computers & Electrical Engineering, vol. 67, pp. 708-728, 2018.

[48] K. a. Z. J. a. Y. K. a. L. C. a. W. C. Hao, "Path Planning of Mobile Robots Based on a Multi-Population Migration Genetic Algorithm," Sensors, vol. 20, no. 20, p. 5873, 2020.

[49] C.C.Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," IEEE Transactions on Systems, Man, and Cybernetics, pp. 404 - 418, vol. 20, no. 2, March-April 1990.

[50] N. P. JOHN YEN, "Path planning and execution using fuzzy logic," 17 Aug 2012. [Online]. Available: https://doi.org/10.2514/6.1991-2801.

[51] L. Meng Wang, "Fuzzy logic based robot path planning in unknown environment," International Conference on Machine Learning and Cybernetics, pp. 813-818 Vol. 2, 2005.

[52] J. Zhu, Theory and application of fuzzy contro, China: Beijing：China Machine Press, 1995, p. 344—355.

[53] A. S. C. RoyGlasius, "Neural Network Dynamics for Path Planning and Obstacle Avoidance," Neural Networks, Volume 8, Issue 1, , pp. 125-133, 1995.

[54] M. M. Simon X. Yang, "Real-time Collision-free Path Planning of Robot Manipulators using Neural Network Approaches," Autonomous Robots 9, pp. 27-39, 2000.

[55] C. L. Simon X. Yang, "A neural network approach to complete coverage path planning," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 1, pp. 718-724, Feb. 2004.

[56] S. X. Y. A. R. W. Z. Y. Hong Qu, "Real-Time Robot Path Planning Based on a Modified Pulse-Coupled Neural Network Model," in IEEE Transactions on Neural Networks, vol. 20, no. 11, pp. 1724-1739, doi: 10.1109/TNN.2009.2029858, Nov. 2009.

[57] Q. J. S. C. Rehder E, "Driving like a human: Imitation learning for path planning using convolutional neural networks," in International Conference on Robotics and Automation Workshops, 2017.

[58] H. W. M. A. E. S. Y. Keyu Wu, "Achieving Real-Time Path Planning in Unknown Environments Through Deep Neural Networks," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3031962., 2020.

[59] F. G. A. R. A. P. Qingbiao Li, "Graph Neural Networks for Decentralized Multi-Robot Path Planning," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11785-11792, doi: 10.1109/IROS45743.2020.9341668, 2021.

[60] X. M. M. S. A. Y. A. F. K. Ben Beklisi Kwame Ayawli, "Optimized RRT-A* Path Planning Method for Mobile Robots in Partially Known Environment," Information Technology and Control, pp. 179-194, 6 4 2019.

[61] X. L. ,. L. ,. H. ,. Z. Xing Lan, "Research on Robot Global Path Planning Based on Improved A-star Ant Colony Algorithm," IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, pp. 613-617, 2021.

[62] J. W. X. S. a. Y. L. Jing Zhang, "Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star," nternational Journal of Advanced Robotic Systems , pp. 1-10, 2021.

[63] O. M. R. S. Ulises Orozco-Rosas, "Mobile robot path planning using membrane evolutionary artificial potential field," Applied Soft Computing, pp. 236-251, 2019.

[64] V. R. K. K. S. R. F. C. S. K. D. P. a. A. M. Sangeetha, "A Fuzzy Gain-Based Dynamic Ant Colony Optimization for Path Planning in Dynamic Environments," Symmetry, 2021.

[65] W. P. N. A. F. C. Abdullah Alomari, "Dynamic Fuzzy-Logic Based Path Planning for Mobility-Assisted Localization in Wireless Sensor Networks," Sensors, 2017, 17(8), 1904. [Online]. Available: https://doi.org/10.3390/s17081904.

[66] C. H.-h. G. W.-k. DU Xin, "Neural network and genetic algorithm based global path planning," Journal of Zhejiang University SCIENCE, 6A(6), pp. 549-554, 2005.