Expert Systems With Applications 227 (2023) 120254

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Path planning techniques for mobile robots: Review and prospect

Lixing Liu, Xu Wang, Xin Yang *, Hongjie Liu, Jianping Li, Pengfei Wang

College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding 071000, China

A R T I C L E I N F O

Keywords:
Path planning
Mobile robots
Key technologies
Algorithms

A B S T R A C T

Mobile robot path planning refers to the design of the safely collision-free path with shortest distance and least time-consuming from the starting point to the end point by a mobile robot autonomously. In this paper, a systematic review of mobile robot path planning techniques is presented. Firstly, path planning is classified into global path planning and local path planning according to the mastery of environmental information. In the global path planning, environment modeling methods and path evaluation method are introduced. The methods of environment modeling include grid method, topology method, geometric feature method and mixed representation method. In the local path planning, we introduce the sensors commonly used in the detection environment, including laser radar and visual sensor. Next, according to the characteristics of algorithms, mobile robot path planning algorithms are divided into three categories: classical algorithms, bionic algorithms and artificial intelligence algorithms. Among the classical algorithms, we introduce the cell decomposition method, sampling based method, graph search algorithm, artificial potential field method and dynamic window method. In the algorithm based on bionics, we introduce genetic algorithm, ant colony algorithm, gray wolf algorithm, etc. in detail. In artificial intelligence algorithm, we introduce neural network algorithm and fuzzy logic. Finally, we compare the key technologies of mobile robot path planning in the form of graphs and charts based on the classification statistics of the collected literature to provide references for future research.

1. Introduction

Mobile robots are intelligent devices that can perform specific tasks independently in complex environments and do not rely on human beings. Today, mobile robots are widely used in many special situations such as deserts, mines, battlefields, and disaster relief (Patle et al., 2019). Especially since the outbreak of the COVID − 19 in 2020, we have become more aware of the convenience brought by mobile robots. Mobile robots can deliver material supplies without assistance or direct contact of human beings, greatly reducing the risk of epidemic transmission while increasing work efficiency.

Mobile robot systems are broadly divided into three modules, including information perception, path planning and motion control. Path planning is the bridge between information perception and motion control, and it is a significantly fundamental part of a mobile robot system. Advanced path planning techniques for mobile robots can reduce capital investment and robot wear and tear. The continuous development of path planning technology has injected fresh vitality into all walks of life. For example, in life, the sweeping robot can take the place of humans to do housework. And driverless cars do not need to drive themselves. As long as the target location is entered, they can provide an optimal route and arrive safely and accurately, subverting the way people used to travel; In production, the warehouse transport robot can independently sort goods and transport them to the designated location, which greatly saves manpower; In emergency rescue and disaster relief, mobile robots can quickly, accurately and safely find targets in the changing dangerous environment; In scientific research, mobile robots can take the place of humans to enter the harsh environment to help humans detect unknown planets and complete the acquisition task; In military science and technology, mobile robots can independently carry out reconnaissance activities on the battlefield to obtain enemy information. These application fields all use path planning technology, so it is of great significance to study path planning technology in the development of the 21st century.

Path planning can generally be divided into global path planning and local path planning according to the level of information about the environment (Mohanty et al.,2021). Global path planning means that the robot is aware of the environment and can reach the target by following a predefined path, based on this feature, global path planning is also called offline path planning or static path planning. Local path

* Corresponding author.
E-mail address: yangxin@hebau.edu.cn (X. Yang).

https://doi.org/10.1016/j.eswa.2023.120254
Received 16 August 2022; Received in revised form 20 April 2023; Accepted 22 April 2023
Available online 3 May 2023

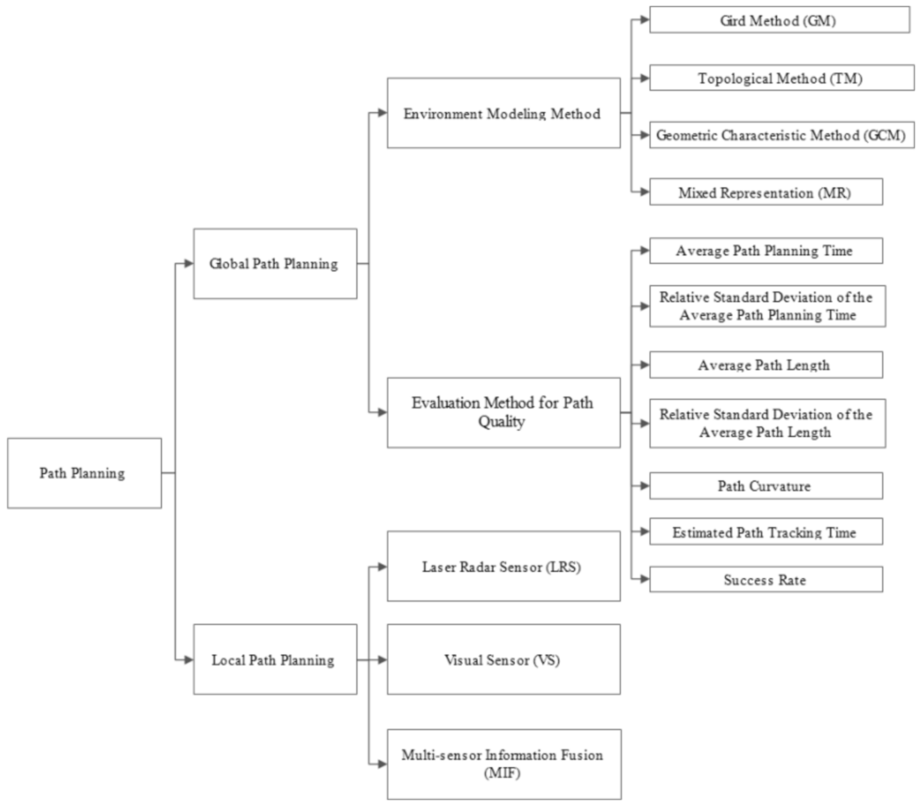


Fig. 1. Key contents of global and local path planning.

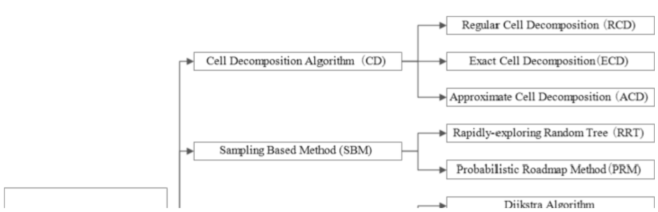


Fig. 2. Algorithm classification block diagram.

planning means that the robot is partially or completely unknown to the environment and carries out real-time monitoring by components and reacts according to the definition, so local path planning is also called online path planning or dynamic path planning. Local path planning is characterized with good flexibility, while the disadvantage being that the planned path may be locally optimal but globally without being guaranteed, or even the target may be unreachable. Global path planning, on the other hand, requires the mobile robot to understand and build a global map model, using a search and seek algorithm on the map model to obtain the optimal or suboptimal path to guide the robot to move safely toward the target point in the actual environment. In practical path planning strategies, it is often necessary to integrate systems that combine global path planning, which aims to find the global optimal path, and local path planning aims to achieve real-time obstacle avoidance planning.

There is no fundamental difference between global path planning and local path planning. Many methods that are applicable to global path planning can be improved for local path planning, while methods that are applicable to local path planning can also be improved for global path planning. The robot can better plan its path from the start to the end if they work together. In global path planning, we focus on the modeling method of the environment and the method for evaluating the merits of the path. In local path planning, we put emphasis on several environmental acquisition devices. See Fig. 1 for details.

Since the 1950s, researchers have proposed many algorithms to solve the optimal path of mobile robots. In this chapter, we analyze the path planning of mobile robot from the perspective of algorithm. According to the characteristics of algorithms, researchers generally divide them into three categories: classical algorithms, bionic algorithms and artificial intelligence algorithms. The detailed classification is shown in Fig. 2.

The rest of this paper is organized as follows. Section 2 summarizes the characteristics and development trends of global path planning and local path planning. Section 3 classifies and analyzes the advantages, disadvantages and development trends of mainstream algorithms from the perspective of algorithms. Section 4 conducts quantitative statistics and analysis on the collected literature, and forecasts the development trend of path planning technology.

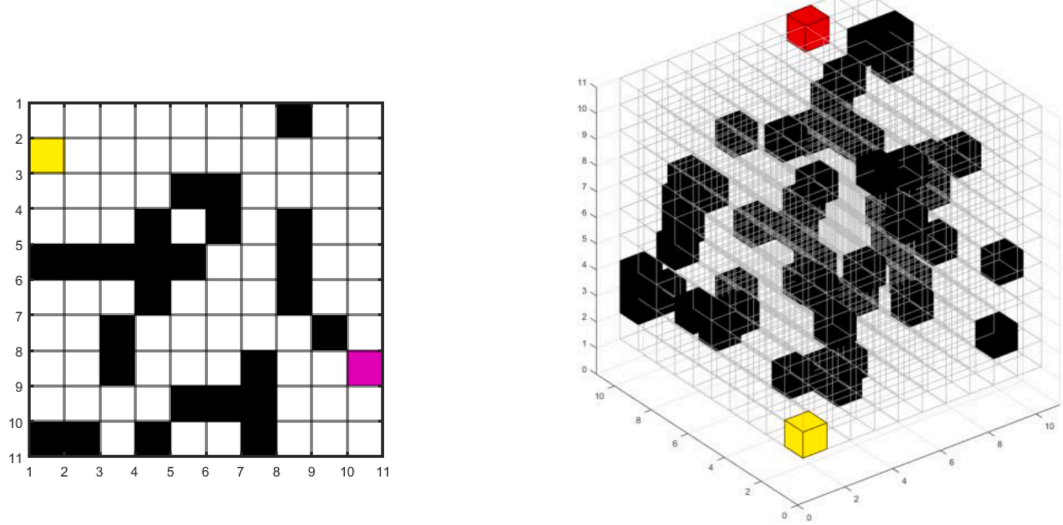
2. Global path planning and local path planning

2.1. Global path planning

Global path planning is to design a safely collision-free path for the robot from the starting point to the end point based on the available environmental information. The accuracy of the planned path depends on the accuracy of the environment information. The process of global path planning can be regarded as finding the optimal solution. Therefore, in the global path planning in this chapter, we will introduce the method of environment modeling and the evaluation method of path quality (Zhang et al., 2018).

2.1.1. Environment modeling

The essence of environment modeling is the organization and utilization of the perceived environment. A proper environment model can effectively help mobile robots to search the path. Therefore, exploring environment models that help robots operate efficiently is a key aspect



（a） 2-D （b） 3-D

Fig. 3. Grid environment.

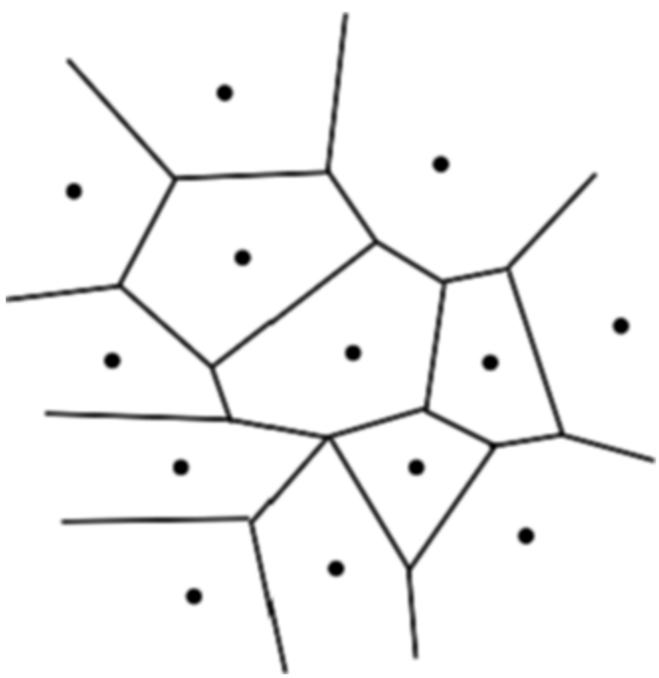


Fig. 4. Voronoi diagram.

of improving robot intelligence and driving robot deployment in a variety of environments. Zou (2005) focused on four common approaches to build environment models, namely, GM, TM, GCM, and MR.

(1) GM (grid method).

Howden (1968) proposed GM, which is essentially a partitioning of a mobile robot's workspace into a series of network cells with binary information and of equal size. When there is no obstacle in a grid, we call it a free grid and the mobile robot can walk freely. When there is an obstacle in a grid, even if the obstacle does not fill the whole area of the grid, researchers generally use the expansion method to expand the obstacle to fill the whole grid and call the grid an obstacle grid. The free grid is usually marked by 0 and the obstacle grid by 1 (as shown in Fig. 3). We generally determine the size of the grid according to the actual size of the robot. If the size of the grid is small, then the environment model will be very clear and the planned paths will be safe. However, it will occupy a lot of storage space of the system and also generate more interference signals, which will lead to a long path planning time. If the grid size is larger, the path planning takes less time and solid real-time performance, but if the environment information is blurred, which is not conducive to path planning. So the determination of grid size is very important for the later path-searching.

The default shape in the Grid Method is square, and the search process mostly uses quadtrees or octrees to represent the workspace. In recent years, researchers have proposed honeycomb grid in order to reduce the time and length of path planning and improve the walking safety of mobile robots. Shao et al. (2021) established a square grid-based map and a honeycomb grid-based map for the problem of uneven step length of traditional grid. After analyzing and comparing the results, it was found that the honeycomb-shaped grid has shorter path length and higher safety when performing path planning, and it has smaller obstacle avoidance angle, smoother path, and less energy consumption by the robot when encountering obstacles. In addition, for the grid method, a 2D occupied grid map creation system based on boundary detection is proposed by Oršulić et al. (2019) for supporting robots in path planning and exploration tasks. A multi-level human-like motion planning method is proposed based on a 2D grid map based on LRF by considering multiple constraints of the robot and the environment to guide the robot to navigate autonomously in an unstructured indoor environment (Zhang et al., 2018a).

GM is characterized by its simplicity and ease of implementation as well as easy extension to 3D environments. A square grid was established for 3D path planning of UAV (unmanned aerial vehicle) (Wu et al., 2021c). The consistency and normality of the grid simplified the adjacency relations in the grid space. In the Chen's literature (Chen et al., 2021), when modeling the environment for a mobile camera, white grids are used to represent the areas where the camera can pass freely and black grids represent the obstacle areas, and other colors represent the starting point or end point of the robot. After giving each grid a passage factor, the path planning problem becomes a problem of finding the optimal path between two grid nodes on the grid network. However, it has certain requirements on the size of the working area. If the area is too large, the number of grids will increase sharply and there is the problem of combinatorial explosion when performing path search. For example, in Fig. 3, we use MATLAB to generate a 10*10 2D environment, which takes 0.096 s, and a 10*10*10 3D environment, which takes 0.557 s.

(2) TM (topological method).

TM is an environmental modeling method proposed by Professor Zhang of Tsinghua University (China) in 1984 (Chien et al. 1984). Its basic idea is to use nodes to represent a particular location and edges to represent the connection of these locations, which can be expressed as $G = (V, E)$ to characterize the free space, where V denotes the set of vertices and E denotes the set of edges connecting the vertices. A Voronoi diagram is usually used to represent the skeleton of the environment feasible region. The basic idea of this method is to generate Voronoi boundaries from the environment model, which has equal shortest distance to each obstacle, and form the vertices of the graph at the junction of the edges.

Voronoi diagrams were proposed by the Russian mathematician G. Voronoi in 1908 and were originally applied to the study of the proximity problem of points in the plane (Preparata et al., 1990). The proximity problem means that if there are certain points in a given plane, and divide the plane according to these points, then any location in the region is closer to the center point than to the other center points (as shown in Fig. 4). In the planning of mobile robots, Voronoi diagrams with obstacles as solid objects are used to describe the networked structure of feasible regions.

TM can effectively reduce the dimensionality of the metric space, Voronoi diagram can convert the two-dimensional metric perceptual information into a network model of the accessible area, making planning a one-dimensional optimal node sequence problem, which is an effective topological representation that cost less storage space and computation of planning time, etc. Voronni diagram represents the state of the neighborhood partitioning centered on obstacles in the environment. Therefore, incremental topological environment modeling can be achieved by modeling sub-regions. Gomez et al. (2020) proposed a global topological representation for the large scale indoor scenes, where each room is modeled as a submap of the topological map for robot navigation and trajectory planning. Yang et al. (2021) presented a new method for estimating the geometry of three-dimensional space using convex polyhedra. Then, the geometric information is used to divide the space into different regions. And these regions are added to the topological map as nodes to guide the search process. Shin and Kim (2021) presented an algorithm to improve the optimality of planning MPP (multi-agent path planning) solutions through parallel pebble diagrams. Embedding the pebble diagram into an arbitrarily complex environment, it is shown that feasibility constraints can be transformed into differentiable geometric constraints and that these constraints are met the requirements of numerical optimization of the constraints. Stenzel et al. (2021) came up with an algorithm to compute a roadmap with intermediate points. The algorithm is able to make full use of as much available space as possible and is able to guarantee the non-overlapping between edges and vertices.

TM path planning is efficient (paths may also be sub-optimal), but it requires little storage space, so it is suitable for applications in large-scale environments. However, topological maps are more difficult to create and maintain, and are prone to confusion when two similar locations exist in the environment.

(3) GCM (geometric characteristic method).

GCM refers to a mobile robot that extracts relatively abstract geometric features from the environment perceived information, such as lines, arcs, and circles, to describe the environment (Guivant et al., 2000). This method builds a relatively compact model and stores a small amount of information, which makes it easy to perform the estimation of position. However, the difficulty of extracting stable features from the raw data that do not change with environmental changes and robot movement leads to the method being mainly applicable to structured operational environments. In addition, finding an exact match between observed features and continuously updated environmental information is a technical difficulty. The shortcoming of the geometric representation is the difficulty of environmental geometric feature extraction, especially in complex unstructured environments. Liang et al. (2018) proposed a path planning method for UAV based on GCM. The method focuses on the shape of obstacles and finally finds a collision-free path. The obstacles are organized in a list of tree. The collision-free path is generated by querying the blocking obstacles and connecting the subgoals of them. Jafarzadeh and Fleming (2018) proposed a new accurate algorithm to solve the path planning problem, which involves finding the shortest collision free path from the starting point to the target point in a 2D environment containing convex and non convex obstacles. A linear network connecting the vertex and the starting point of the obstacle is constructed, which is smaller than the network generated by the visibility map. Then find the shortest path from the beginning to the target point in the network.

(4) MR (mixed representation).

In 1998, Thrun (1998) proposed a method for extracting topological features from global metric maps, using distance information from sonar sensors to extract features during robot navigation. A new node is generated when the distance information of the sonar happens to be significantly different from the features of the known nodes. If the feature information is similar to the features of the known node, a probabilistic localization is performed using a partially observable Markov decision-making method and a topological map of the environment is generated at the same time. However, generating a topology map in regions where the obstacle distance features are not significant enough is still not easy and also sensitive to dynamic environment with sensor noise. In 1999, Yeap & Jefferies (1999) proposed a method to extract a global topology map from a local metric map. The local maps are represented by grids, using a topology-like edge set to connect multiple local maps that already exist. Since the local metric maps are maintained in a small range, the errors can be neglected. Two levels of local and global planning are used in path planning, i.e., regional planning based on occupying grids and global planning based on topological connectivity relations, which is the most primitive Mixed Representation.

MR can combine the efficiency of topology path planning with higher modeling accuracy in metric space to achieve multi-level planning. A topology-metric map construction method for efficient item search is proposed in the study of home-service oriented robots to enable mobile robots to quickly locate target items in complex home environments (Zhang, 2021), providing an important guarantee for robots to efficiently perform domestic operations. Unlike the traditional topology-metric map, in this method, the topological nodes on the map are created by referring to the environmental items, which reduces the redundant nodes unrelated to the item search with lower searching cost of the mobile robot, and improves the path planning efficiency. Gregorio & Stefano (2017) proposed SkiMap, a multi-level environment model framework for robot navigation. SkiMap contains a 3D voxel grid map, a 2.5D height map, and a 2D occupancy grid map, and supports continuous updates to improve the accuracy of the map during walking.

2.1.2. Evaluation method for path quality

The main task in the path search problem for mobile robots is to find the correct search strategy. The search strategies, i.e., path planning algorithms will be analyzed in detail in Section 3. In order to evaluate the capability of each search strategy and measure their performance in an objective way, in this section seven different evaluation method will be presented (Tsardoulias et al., 2016).

(1) Average path planning time - t_m. The average path planning time spent by an algorithm, this criterion is crucial because whatever algorithm must plan the path for the mobile robot efficiently and give results in the shortest possible time.

(2) Relative standard deviation of the average path planning time. The relative standard deviation of the average path planning time spent by an algorithm, RT, the calculation formula is,

Table 1
Summary of LRS basic parameters.

Name	Distance (m)	Channel	Distanceerror (cm)	Response time (ms)	Measurement frequency (Hz)	Horiz FOV (°)	Horiz resolution (°)	Vertical FOV (°)	Vertical resolution (°)	Weight (g)
Velodyne PUCK VLP-16	100	16	±3	–	5 ~ 20	360	0.1 ~ 0.4	−15 ~ +15	2.0	830
Velodyne ULTRA PUCK VLP-32C	200	32	±3	–	5 ~ 20	360	0.1 ~ 0.4	−25 ~ +15	0.33	925
SickRS1104C-111010	64	4	±6	–	5 ~ 15	275	0.25	−7.5 ~ +7.5	0.25	1200
SickRS1104C-111011	64	4	±6	40	5 ~ 15	275	0.25	−7.5 ~ +7.5	0.25	1200
Ouster OS-1 16	120	16	±1.5	10	10 ~ 20	360	±0.01	−15.8 ~ +15.8	±0.01	330
Ouster OS-1 64	120	64	±3	–	10 ~ 20	360	0.09	−15.8 ~ +15.8	0.52	330
LeiShen Ls Lidar C16 3D RB-Lsl-01	150	16	±3	–	5 ~ 20	360	0.1	−15 ~ +15	2	1500
LeiShen Ls Lidar C16 RB-Lsl-10	50	16	±7	–	5 ~ 20	360	0.1	−15 ~ +15	2	1500
Robosense RS LIDAR 16	150	16	±2	–	5 ~ 20	360	0.1/0.2/0.4	−15 ~ +15	2	840
Robosense RS LIDAR 32	200	32	±3	–	5 ~ 20	360	0.1/0.2/0.4	−15 ~ +25	0.33	1000
Robosense RS Ruby	200	128	±3	–	10 ~ 20	360	0.2/0.4	−15 ~ +25	0.1	3750
RPLIDAR A2	18	1	–	–	5 ~ 15	360	–	0	–	190
RPLIDAR A3	25	1	±5	–	10 ~ 20	360	0.225	0	0.225	190
Sick TIM551	8	1	±6	67	15	270	1	0	1	250
Hokuyo PBS-03JN	6	1	–	180	–	178	1.8	0	1.8	500
Hokuyo URG-04LX-UG01	5.6	1	±3	100	10	240	0.36	0	0.36	160
Sick LMC132-11101 VdS	18	1	±3	20	50	270	0.25 ~ 0.5	0	0.25 ~ 0.5	1100
Sick LMS111	20	1	±3	20	25 ~ 50	270	0.25 ~ 0.5	0	0.25 ~ 0.5	1100
Sick LD-MRS400001	300	4	±30	–	12.5 ~ 50	85	0.125 ~ 0.5	0	0.125 ~ 0.5	1000
SickRS1104C-111010	64	4	±6	–	5 ~ 15	275	0.25	−7.5 ~ +7.5	0.25	1200
SickRS1104C-111011	64	4	±6	40	5 ~ 15	275	0.25	−7.5 ~ +7.5	0.25	1200
Sick LD-MRS400001	30	4	±30	–	12.5 ~ 50	85	0.125 ~ 0.5	0	0.125 ~ 0.5	1000
Sick LMC132-11101 VdS	18	1	±3	20	50	270	0.25 ~ 0.5	0	0.25 ~ 0.5	1000
Sick TIM551	8	1	±6	67	15	270	–	0	–	250

Here "-" represents unknown and FOV means field of view. It should be noted that the channel refers to the number of laser transceiver modules in the LRS. Single-line LRS can only obtain two-dimensional plane information, while multi-line LRS can obtain three-dimensional information. The more channels, the higher the comprehensive performance of the LRS, but more laser transceiver modules will also lead to the large volume and high cost of the LRS.

$$RT = \frac{\sqrt{\frac{1}{N\bullet K}\bullet\sum_{i=1}^{N\bullet K}(t_i - t_m)^2}}{t_m} \tag{1}$$

In the formula, t_i refers to the time spent by an algorithm in planning the path in the ith time; N refers to the target points in the environment; and K refers to the times of the algorithm is executed. Relative standard deviation is a metric in statistics that reflects the degree of accuracy and repetition. From the perspective of algorithm planning time, the relative standard deviation represents the reliability of an algorithm, that is, the degree of deviation of the time of each run of the algorithm from the average time.

(3) Average path length. The average length of a path generated after an algorithm run - D_m. The average path length is directly related to the path tracking time of a mobile robot, so this evaluation criterion is essential. We assume that each path has many target points and mark the ith target point as $Path_i$. in addition, the path length of the robot to reach the target point $PathSize=|Path|$. So the formula for D_m is,

$$D_m = \sum_{i=1}^{PathSize} Dist(Path, Path_{i+1}) \tag{2}$$

The formula $Dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$ refers to the Euclidean distance between the point p and the point q on the coordinate axis.

(4) Relative standard deviation of path length - RD. The standard deviation of path length after the calculation of an algorithm, similar to the relative standard deviation of execution time. The calculation formula is,

$$RD = \frac{\sqrt{\frac{1}{N\bullet K}\bullet\sum_{i=1}^{N\bullet K}(D_i - D_m)^2}}{D_m} \tag{3}$$

In the formula, D_i refers to the path length obtained by running an algorithm for the ith time. This criterion assesses the accuracy of the paths generated by an algorithm, i.e., whether the algorithm generates the same paths in different map environments.

Path curvature - RC. path curvature reflects the smoothness of the path generated by the algorithm, i.e. the amount of rotational motion that the robot must perform when tracking the path. Here, according to the literature (Tsardoulias et al., 2016), we specify that the RC is bounded in the range [0,1]. The turning motion of the robot is measured as it crosses a specific path. To simplify the computation, the default vehicle turns only at the target point and the purpose of the turn is to align with the next target point. What's more, it is assumed that the robot always rotates the smallest angle when turning, so the angle range of the turn is [0°,180°]. Obviously, the best case is that the robot is aligned with the final target and no rotational motion is required, so the angle sum is 0°. On the other hand, the worst case is to perform a 180° turn on each sub-target,

Table 2
Performance parameters of common VS.

Name	TS	Pixel	PS (μm)	QE (%)	DN (e)	SNR (dB)	AST (γ)	DR(dB)
IMX428	1.1″	3208*2200	4.5	51.79	2.77	40.44	5.38	70.59
IMX287	1/2.9″	720*542	6.9	61.82	3.80	43.51	7.55	74.43
IMX392	1/2.3″	1920*1200	3.45	60	2.38	40.32	4.81	71.45
IMX265	1/1.8″	2048*1536	3.45	59.66	2.31	40.30	4.93	71.74
AR0134	1/3″	1280*960	3.75	57	5.12	37.49	9.73	59.97

Here TS means target size; PS means pixel size; QE means quantum efficiency; DN means dark noise; SNR means signal-to-noise ratio; AST means absolute sensitivity threshold; DR means dynamic range.

which is unrealistic, but from this it is possible to calculate the upper limit of *RC*, which is calculated as follows.

$$RC = \frac{\sum_{i=1}^{PathSize}\theta_i}{(PathSize - 1)\bullet 180^{\circ}} \in [0, 1] \tag{4}$$

(6) Estimated path tracking time - *ETT*. *ETT* refers to the time spent by an algorithm-generated path robot in performing path tracking. Obviously, the time calculated by this criterion includes the time spent by D_m and the rotational motion that the vehicle must perform. The following assumptions are made to simplify the calculation a) the vehicle performs a purely linear or rotational motion, and not a combination of both; b) the linear velocity is constant, denoted LS m/sec, and the angular velocity is constant, denoted RS rads/sec; c) the transition from linear to rotational motion is instantaneous and vice versa, d) no friction or slippage occurs during each movement; e) the occupied grid cell size (in meters) is denoted as *OGCD*, and the accumulated rotational movement is denoted as $AS = \sum_{i=0}^{PathSize-1}\theta_i$. The resulting *ETT* is calculated as follows.

$$ETT = \frac{D_m \bullet OGCD}{LS} + \frac{AS}{RS} \tag{5}$$

(7) Success rate - *SR*. *SR* is the most important measure of a path planning algorithm and refers to the success rate of an algorithm planning a path that can reach the target point safely and collision-free, i.e., its ability to compute the path regardless of a) environmental conditions and b) the pose of the robot and the target. Path planning algorithms can fail due to two main reasons a) due to violation of execution time constraints and b) due to environmental constraints, i.e., narrow passages or lack of traversable areas. As mentioned above, the number of executions of each method in an environment is $N\bullet K$. Therefore, by noting S as the number of successfully created paths, the success rate of each algorithm in a given environment is calculated as follows.

$$SR = \frac{S}{N \bullet K} \bullet 100\% \tag{6}$$

2.2. Local path planning

The difference between local path planning and global path planning is that the environmental information faced by the mobile robot is partially or completely unknown. Therefore, one of the focuses of local path planning is the environment detection device, and the current open source mapping algorithms rely on two types of sensors, that is, LiDAR sensors and vision sensors. In addition, sensor fusion as a starting point to build a highly stable system with multi-sensor combinations is also an inevitable trend of improving mobile robots (Nian, 2021).

2.2.1. LRS (laser radar sensor)

LRS is frequently used in mobile robots for map building, before that laser echo-based laser rangefinders were the main instrument for indoor engineering distance measurement. Its reliability is higher with more mature development, and its robustness to light environment is also higher, but it is more expensive, and it is LRS that is configured in the sweeping robots for home cleaning. Laser ranging is one of the earliest applications of laser technology, and laser ranging is characterized by long detection distance and high measurement accuracy. Laser ranging methods are mainly divided into two categories, one is the pulse ranging method; the other is the continuous wave ranging method (An & Zeng, 2001). LRS is similar to laser rangefinder in that it uses environmental geometric information as feedback as well as environmental portrayal, with small feature changes under illumination changes as well as viewpoint changes, dense feedback information, and stable perception of the exterior (Chen, 2020). Therefore, it has been applied in many studies. The classical work on path planning for mobile robots based on LRS, including the book *Probability Robotic* by Thrun (2002) and the AMCL open source localization algorithm package, have a considerable application base.

LRS is a complex opto electromechanical system in essence. Its advantages and disadvantages cannot be determined by a single index. It needs to be comprehensively determined by combining actual application scenarios and corresponding functions. In order to facilitate the reader's choice of lidar, we collected some information on the website and listed some basic parameters of LRS on the market in Table 1.

The geometric structure information of laser feedback is for different sensing modes of the same environment. For the problem of difficult convergence of laser localization under similar structural features, its difficulty in environmental modeling work in the face of outdoor open environments, and environments with highly repetitive geometric information, often due to the high repetition of environmental sensing information and the single content of feedback information.

2.2.2. VS (visual sensor)

VS is a typical passive sensor, in which the sensing information is obtained from the reflection of light from the surface of the object. Compared with LRS, VS has a wider detection range and can acquire rich image information. Depending on the number of VS and their functions, they can be classified into monocular vision detection, stereo vision detection, and omnidirectional vision detection (Liu, 2018). In recent years, obstacle detection methods based on omnidirectional vision have been widely studied due to the advantages such as wide perception range of omnidirectional vision system. Shi et al. (2017) built an omnidirectional vision system on a humanoid robot to achieve real-time environment perception and obstacle detection in 360°. A feature point extraction system based on deep convolutional neural networks was proposed by Daniel et al. in 2017, using a dual-network architecture that first extracts salient feature points and then computes point locations, which is a real-time system but more streamlined and can easily run on a single CPU (Detone et al., 2017). In 2022, Chang et al. (2022) proposed a Yolov4 micro-network-based visual mapping algorithm. Meanwhile, a dynamic feature point elimination strategy based on traditional ORB-SLAM is proposed. In order to obtain semantic information, target detection is performed while extracting feature points of the image. In addition, the polar line geometry algorithm and LK optical flow method are used to detect dynamic targets. Dynamic feature points are removed in the tracking thread and only static feature points are used to estimate the camera position. After evaluation on the TUM dataset, this method is featured by well real-time performance.

Since the VS carried by the mobile robot is not directly provided to the human eye for observation, but used for environmental modeling, multiple parameters need to be considered when evaluating its image quality. To facilitate readers' choice of VS, Table 2 lists the performance parameters of common VS.

However, vision-based environment modeling still faces many problems, mainly: (1) when using vision sensors for environment detection, due to uneven lighting and other factors, there may be object shadow interference in the environment, which may easily cause obstacle miss detection, wrong detection and other situations, making the subsequent construction of the navigation map lack of reliability, how to adaptively and efficiently remove environmental shadow interference based on environmental brightness information, so as to enhance the reliability of environmental perception information, has important research significance. (2) Binocular vision system can recover parallax based on 2D image information and use triangulation principle to get 3D information in the scene, but the three-dimensional matching link is of low accuracy and serious time consuming, so it is important research value to improve the accuracy and real-time of binocular vision measurement. (3) Mobile robot navigation tasks require target localization, static target position acquisition is relatively simple and can be set in the map in advance, dynamic targets have motion characteristics, there will be background interference, partial occlusion, target rotation and rapid motion blur and other challenging scenarios, how to design dynamic target tracking algorithms with real-time, robustness and accuracy has a greater challenge. (4) VS-based path planning, because the target is moving in real time, a more concise and real-time dynamic path planning algorithm is needed, and further research is needed on how to fuse multiple vision sensors information and combine image constraints and spatial location constraints to better plan the path.

2.2.3. MIF (multi-sensor information fusion)

MIF refers to the fusion of data from multiple sensors by some algorithm for mobile robots in complex, dynamic and uncertain environments, so as to make up for the deficiencies of a single sensor in some aspects and reduce the redundancy of a single sensing information, in order to make full use of the information from multiple sensors, so that the system can more reliably and accurately reflect the characteristics of the external environment to which the robot is exposed in a shorter period of time at a smaller cost (Chen, 2020).

According to the above analysis of the pros and cons of LiDAR sensors and computer vision sensors, LiDAR has high perceptual stability in small indoor environments, but in large environments, the cumulative error and the lack of closed-loop detection make the robot build maps with low accuracy. On the other hand, scene recognition localization in complex environments is not possible due to the small amount of information collected by sensors. Computer vision modeling techniques with low data acquisition difficulty and large amount of information are widely favored by researchers. However, the large amount of two-dimensional image information acquired by vision sensors can be used for the implementation of closed-loop detection. However, vision solutions lack the perception of distance, while laser sensing happens to have a unique advantage in this area (Liang, 2016). Due to the complexity of the working environment and the uncertainty of the robot's own state, it is difficult to perceive and model the external environment with just one sensor. To solve the problems of low information and poor robustness of single sensor, the fusion technology of multi-sensor information has received more and more attention and research from scholars, making it a feasible method to improve the environment perception of robot.

There are several types of solutions for sensor fusion, the more common one is the fusion of vision and inertial navigation, which requires the help of an inertial measurement unit (IMU), which measures the angular and linear velocities of moving objects and is a powerful sensor tool to calculate the positional attitude through the inter-frame velocity transformation of the robot (Zhang, 2013). However, the poses acquired by the inertial measurement unit alone can have significant drift, and the image information acquired with the camera can be well constrained to calculate and acquire more accurate positional data. The earliest visual-inertial odometry (VIO) scheme was proposed by Mourikis in 2007 as a Kalman filter-based real-time vision-assisted inertial measurement unit navigation algorithm (Mourikis & Roumeliotis, 2007). After this, in 2014, Dr. Li from the University of California proposed improved algorithms for fusion of vision sensors with IMUs in his dissertation, and the paper used some small, inexpensive systems with more limited sensing resources to do motion tracking and develop new commercial opportunities (Li, 2014). In addition to vision and inertial guidance fusion solutions, there are also a number of excellent solutions in the field of laser and vision fusion. V-LOAM is the system that excels in all aspects and was proposed by Zhang's group in 2015. He is a self-motion estimation method combining monocular camera's and 3D LiDAR, capable of accurately estimating the motion of 6 degrees of freedom as well as the metric representation of the environment space, among others (Zhang & Singh, 2015). The entire system is divided into two sequentially interleaved processes, one of which uses a visual odometer at 60 Hz for motion attitude estimation; the other uses LiDAR at a low frequency of 1 Hz to assist in improving motion estimation and eliminating the cumulative error and point cloud distortion introduced by the visual odometer, a method that does not require closed-loop detection and corroborates the accurate estimation of the odometer during map building. The LIMO algorithm proposed by Johannes and Alexander et al. in 2018 is also typical for laser vision fusion, mainly applied in autonomous driving, but does not use camera sensors to acquire images in real time, but uses images, LIDAR to acquire depth information for RGB images, and the optical flow method to estimate camera motion, and the semantic method for the annotation of some landmark points, which is also able to determine the weights of target points and remove outliers at the same time. The method has been open-sourced and has received high rankings on the KITTI dataset for large scale applications in mobile robots (Graeter et al., 2019).

For mobile machines applied to simple scenarios, 2D flat environment models are very efficient, which can accurately scan the distance of obstacles for the robot and accomplish real-time navigation tasks. For mobile robots in complex scenarios, 2D maps are no longer sufficient. From the viewpoint of security and practicality, it is necessary to construct 3D maps. For the previously introduced LiDAR sensors and vision sensors, combined with the related research on multi-sensor fusion mobile robot environment modeling by scholars at home and abroad, the complementary sensing capabilities of multiple sensors and the application of information fusion methods can not only make the mobile robot have the ability to perceive the 3D environment, but also the ability of mutual noise suppression between sensors to improve the robot sensing accuracy, making the acquired information more accurate and effective (Su et al., 2017c).

Mohanmadhossein investigated the close coupling of vision and LIDAR applied to 3D environment perception for mobile robots (Daraei, 2018), mainly including LIDAR and image temporal inconsistency during LIDAR and vision fusion, and studied the modeling of occlusion due to parallax during the fusion of both information. LIC-Fusion proposed an efficient tightly coupled multimodal laser-inertial-guidance-visual odometry fusion method within the framework of MSCKF (Zuo et al., 2019; Ma et al., 2019), which can achieve robust self-motion estimation under different environments and strenuous motions. Chinese university research institutes and others have also carried out various researches and achieved some results. Meanwhile, a number of traditionally renowned schools and research institutes in robotics, mechanics, and automation in China, such as Tsinghua University, Harbin Institute of Technology, and Shenyang Institute of Automation, have made a series of achievements in multi-sensing fusion algorithms and corresponding hardware and software development (Ding et al., 2016; Zhao et al., 2017). There is a promising trend and huge market of this research both at home and abroad.

However, the development of sensor fusion schemes is relatively short and there are still many research results to be explored, and there

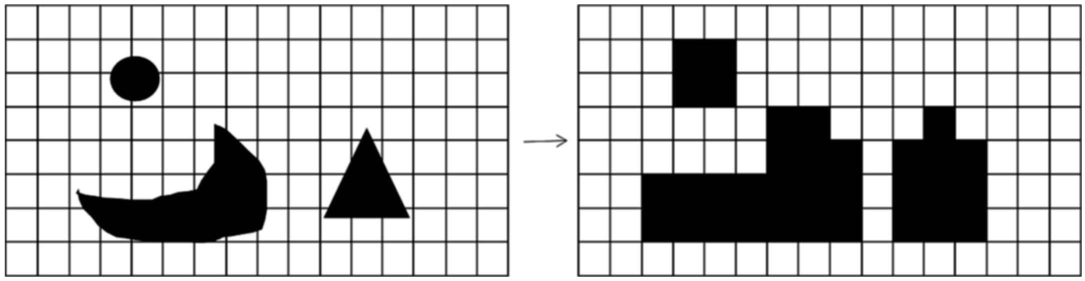


Fig. 5. Schematic diagram of the RCD.

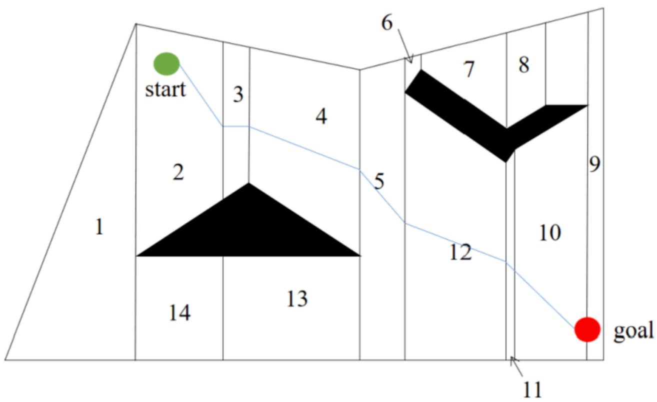


Fig. 6. Schematic diagram of the ECD.

is no more mature scheme in this field for complex indoor environment mapping. The fusion algorithm does not provide a closed-loop detection module (Zhang & Singh, 2015), so the accuracy of the mapping needs to be improved, and the algorithm requires high hardware equipment, which is not fully applicable to small and lightweight indoor service-oriented robots; some dense visual point cloud mapping can greatly restore Some dense visual point cloud mapping can greatly recreate indoor scenes, but the computation is huge and usually needs to be accelerated by cuda and GPU, which consumes more computational resources in large scale scenes and is difficult to run well in embedded systems.

3. Path planning algorithm

Since the 1950s, many algorithms have been proposed by researchers to seek the optimal path of a mobile robot. In this chapter, we analyze mobile robot path planning from the perspective of algorithms. According to the characteristics of the algorithms, researchers generally classify them into three categories: classical algorithms, bionic algorithms and artificial intelligence algorithms. The classical algorithms including cell decomposition method, sampling based method, graph search algorithm, artificial potential field, and dynamic window approach. Among the bionic algorithms include genetic algorithm, ant colony optimization algorithm, particle swarm algorithm, firefly algorithm, bacterial foraging algorithm, cuckoo search algorithm and artificial bee colony algorithm. Artificial intelligence algorithms mainly include fuzzy control algorithms and neural network algorithms.

3.1. The classical algorithm

Initially the classical path planning algorithms are very popular in the field of autonomous navigation for mobile robots because they all have a common advantage, easily observed computation results. In contrast, classical algorithms are more suitable for application in global path planning with a deterministic environment, but of course they can be made applicable to uncertain environments by different optimization methods, but some researchers are prone to generate problems such as high computation cost and low computation efficiency.

3.1.1. CD (cell decomposition approach)

CD first utilizes the grid method introduced in Section 2 to divide the configuration space in which the mobile robot is located into a free grid and an obstacle grid, and uses a connectivity map to traverse from one cell to the next, forming a collision-free path from the starting point to the target point (Šeda, 2007). The cell decomposition approach is classified into regular cell decomposition algorithm, approximate cell decomposition algorithm, and exact cell decomposition algorithm based on the strategy of dividing grids.

(1) RCD (regular cell decomposition).

RCD commonly pre-sets the size of the cell and uses a rectangular shape, thus dividing the environment of the mobile robot into small cells of the same shape and equal size. For the safety of the mobile robot, we also use inflation to enlarge the obstacles, i.e., whenever there is an obstacle in a small cell, no matter how large it occupies, this cell is classified as an obstacle cell (as shown in Fig. 5). By counting the number of collected literature, the regular cellular decomposition algorithm is the most common compared to the other two cellular decomposition algorithms, which can plan paths for mobile robots well in environments without localized minimal obstacles, which is very applicable to drones and unmanned vehicles. Bao et al. (2022) added A* algorithm with artificial potential field method for unmanned path planning under the shaft of a trackless robot based on RCD. Zhou et al. (2018) proposed a variable cell strategy based on RCD applied to the path planning of UAVs. Challita et al. (2018) incorporated a deep learning algorithm in a honeycomb-shaped RCD for UAV path planning during obstacle avoidance.

Although RCD is very applicable, it still has some drawbacks. On the one hand, when a barrier is much smaller than the cell, it will still make the whole cell become a barrier cell, this may lead to a path scheme that is not optimal for system planning. On the other hand, if the cell is set too small, it will lead to too much computation and become less efficient (Debnath et al., 2021).

(2) ECD (exact cell decomposition).

The shape and size of cells in ECD are indeterminate; they are determined by the obstacle and the environment. The free space is first decomposed into multiple non-overlapping small cells, which are usually rectangular, trapezoidal or triangular. Then give each small cell a number. These small cells are precisely equal to the original space composed of free cells (as shown in Fig. 6). For the case where the boundary of the obstacle in the environmental space is curved, Jung et al. (2019) proposed an exact cell decomposition algorithm based on opposite angle (OAECD) based on ECD. OAECD algorithm is also focused on the static environment and has poor robustness for dynamic environment. In the Gonzalez's literature the performance of the conventional algorithm was improved by changing the cell decomposition, the graphical weights of small cells and the numbering order (Gonzalez et al., 2017). Park et al. (2015) described a method to represent the generation of local path using the ECD method.

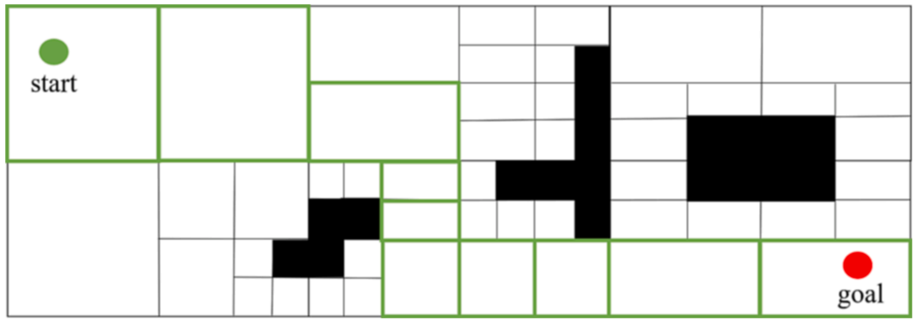


Fig. 7. Schematic diagram of the ACD.

Table 3
Comparison of the different CD algorithms.

Method	Optimal Path	Real time	Memory	Safety
RCD	N	N	N	Y
ECD	N	N	Y	N
ACD	N	N	Y	Y

Here Y stands for “Yes” and N stands for “No”.

Table 4
The pseudo-code of the RRT’s algorithm.

Input: M,Q_{init},Q_{goal}
Result: A path Γ from Q_{init} to Q_{goal}

Γ.init();
for $i = 1$ to n **do**
  $Q_{rand} \leftarrow Sample(M)$;
  $Q_{near} \leftarrow Near(Q_{rand},\Gamma)$;
  $Q_{new} \leftarrow Steer(Q_{rand},Q_{near},StepSize)$;
  $E_i \leftarrow Edge(Q_{new},Q_{near})$;
  if $CollisionFree(M,E_i)$ **then**
    $\Gamma.addNode(Q_{new})$;
    $\Gamma.addEdge(E_i)$;
  end if
  if $Q_{new} = Q_{goal}$ **then**
    Success();
  end if
end for

Here M stands the environment map, Q_{init} stands the starting point of the mobile robot, and Q_{goal} stands the target point.

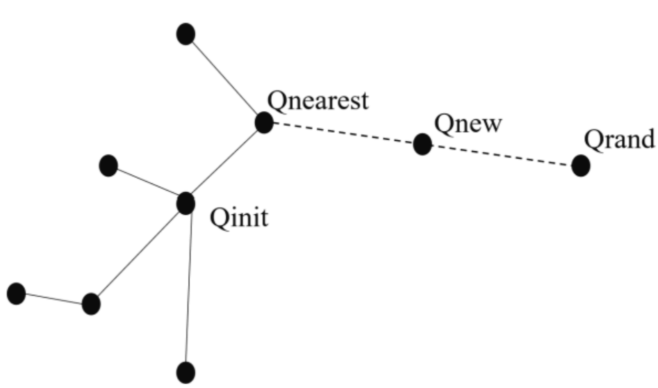


Fig. 8. Schematic diagram of the RRT algorithm.

(3) ACD (approximate cell decomposition).

All small cells of the ACD are of a predetermined shape, usually rectangular. The whole environment is divided into several larger rectangles, each of which is continuous with the other, typically by the “quadtree” method and the “octree” method, and if the large rectangle contains obstacles or boundaries, it is divided into four small cells. This division is performed for all slightly larger cells, and the procedure is then repeated among the smaller cells formed within the final boundaries of the division until the limits of the solution are reached (as shown in Fig. 7).

Guruprasad & Ranjitha (2021) applied ACD to the field of full coverage of mobile robot paths. Samaniego et al. (2019) proposed Recursive Rewarding Modified Approximate Cell Decomposition (RRMACD) to construct small cubic cells in the 3D environment in which the UAV is located, and considers various constraints such as UAV maneuverability, geometry, static and dynamic environmental obstacles, without invoking other algorithms to reduce the calculation and complexity.

The sum space of the small cells formed by ACD does not match the free space of the original robot, and the amount of available space around the generated path is controlled by setting the size of the smallest cell. The free space of the approximate cell decomposition algorithm is locally adapted to the shape of the obstacle in the vicinity of the obstacle, and the degree of adaptation is related to the predetermined minimum cell size. Thus, the minimum cell size setting is crucial. Therefore, most methods operate in a hierarchical manner by generating an initial coarse decomposition and then locally refining that decomposition until a free path is found or the decomposition becomes too meticulous to continue (Latombe, 1991).

To sum up, CD is intuitive and can ensure the minimum local collision probability of mobile robot. However, if the cells formed are relatively rough, CD cannot guarantee that the path is optimal. On the contrary, for RCD and ACD, if the cell division is too careful, it will consume more computing time. For RCD, when an obstacle is far smaller than the cell size, according to the traditional division scheme, it will still cause the entire cell to become an obstacle cell, which may lead to that the path planned is not optimal, and even cannot guarantee to find an existing path. If we divide this cell into free cells, the length of the path may be shortened, but the completeness of the planned path cannot be guaranteed. In order to visually compare CD algorithms, we have done so in Table 3 (Debnath et al., 2021).

3.1.2. SBM (sampling based method)

(1) The RRT (rapidly-exploring random tree) algorithm and the optimization algorithm based on it.

The RRT is an active sampling-based algorithm proposed by LaValle in 1998. The method is suitable for solving path planning problems with different motion constraints and has the ability to handle multi-degree-of-freedom problems. The principle of the RRT algorithm is to quickly search the bit-shaped space nodes to generate a path connecting the starting and target nodes (as shown in Table 4 and Fig. 8) (Abdallaoui et al., 2022).

Usually, researchers can set an upper limit on the running time or an upper limit on the number of searches in order to make the algorithm manageable. If the target point cannot be reached within the limited times, the algorithm returns will fail. In the Palmieri’s literature, a combination of (discrete) arbitrary angle search and (continuous) RRT

Table 5
The pseudo-code of the RRT - Connect's algorithm.

Input: M, Q_{init}, Q_{goal}
Result: A path Γ from Q_{init} to Q_{goal}

Γ_a.init(); Γ_b.init();
for $k = 1$ to K **do**
  $Q_{rand} \leftarrow Sample(M)$;
  $Q_{rand}' \leftarrow Sample(M)$;
  $Q_{near} \leftarrow Near(Q_{rand}, \Gamma_a)$;
  $Q_{near}' \leftarrow Near(Q_{rand}, \Gamma_b)$;
  $Q_{new} \leftarrow Steer(Q_{rand}, Q_{near}, StepSize)$;
  $Q_{new}' \leftarrow Steer(Q_{rand}', Q_{near}', StepSize)$;
  if $CollisionFree(M, E_i)$ **then**
    $\Gamma.addNode(Q_{new})$;
    $\Gamma.addNode(Q_{new}')$;
    $\Gamma.addEdge(E_i)$;
  end if
  if $Q_{new}' = Q_{new}$ **then**
    Success();
  end if
end for

Here M, Q_{init}, Q_{goal} have the same meaning as Table 4.

motion planning is applied to a small time controllable Nonholonomic Wheeled Mobile Robot (Palmieri et al., 2016).

The RRT algorithm can only find the path, but it cannot guarantee that the path is optimal. And it is a purely random search algorithm insensitive to the environment, and when the environment contains a large number of obstacles or narrow passages, the convergence of the algorithm is slow and the efficiency drops dramatically. Next, we present several classical solutions for optimizing the RRT algorithm. The first method is the RRT - Connect algorithm, based on the blindness of the RRT search space, the shortcomings of the node expansion link lack of memory, in order to improve the search speed in space. The RRT algorithm is coupled with a two-tree bi-directional exploration guidance strategy, and a greedy strategy is added to the growth method to speed up the searching and reduce the invalid search of the blank area, and save time (as shown in Table 5 and Fig. 9).

A flexible multi-directional fast exploration tree generation method was used in the Qian's literature (Qian et al., 2020). Based on the principle that trees grow centripetally, new multi-directional trees will be constructed as needed to achieve a specific coverage of space. Then, based on the previous path exploration vertices, the complete paths are formed and optimized by the tree fusion method, and finally the local optimal paths are generated. Experimental results show that the method can effectively improve the search efficiency at a low computational cost, ensuring that each sampled point is not useless information and minimizing the search times of the whole map. the RRT-Connect algorithm adds heuristic strategies, and greedy ideas, but the common drawback of both the RRT algorithm and the RRT-Connect algorithm is that neither of their paths is optimal. No function is added to evaluate the path cost, and the search path strategies are both based on random sampling of the search.

The asymptotically optimal RRT* algorithm, which improves the parent node selection on the original RRT algorithm, uses a cost function to select the node with the smallest cost in the domain of the expanded nodes as the parent, and at the same time, reconnects the nodes on the existing tree after each iteration, thus ensuring the computational complexity and the asymptotically optimal solution (Karaman & Frazzoli, 2011). Its principle is the same process as RTT before finding the point Qnew, and there are two stages after finding Qnew. (as shown in Table 6 and Fig. 10) After these two steps, the mobile robot reduces unnecessary search and storage, which largely improves the search

Table 6
The pseudo-code of the RRT*'s algorithm.

Input: M, Q_{init}, Q_{goal}
Result: A path Γ from Q_{init} to Q_{goal}

Γ.init();
for $i = 1$ to n **do**
  $Q_{rand} \leftarrow Sample(M)$;
  $Q_{near} \leftarrow Near(Q_{rand}, \Gamma)$;
  $Q_{new} \leftarrow Steer(Q_{rand}, Q_{near}, StepSize)$;
  if $CollisionFree(q_{new})$ **then**
    $Q_{near} \leftarrow N_{ear}C(\Gamma, q_{new})$;
    $q_{min} \leftarrow ChooseParent(Q_{near}, q_{near}, q_{new})$;
    $\Gamma.addNodEdge(Q_{new}, q_{new})$;
    $\Gamma.rewire()$;
  end if
end for

Here M, Q_{init}, Q_{goal} have the same meaning as Table 4.

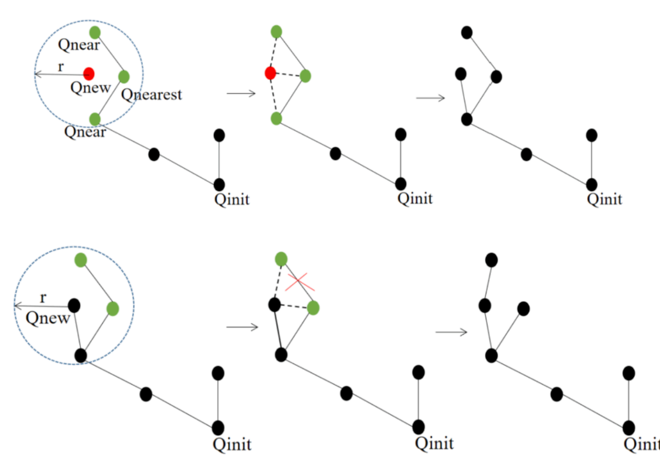


Fig. 10. Schematic diagram of the RRT* algorithm.

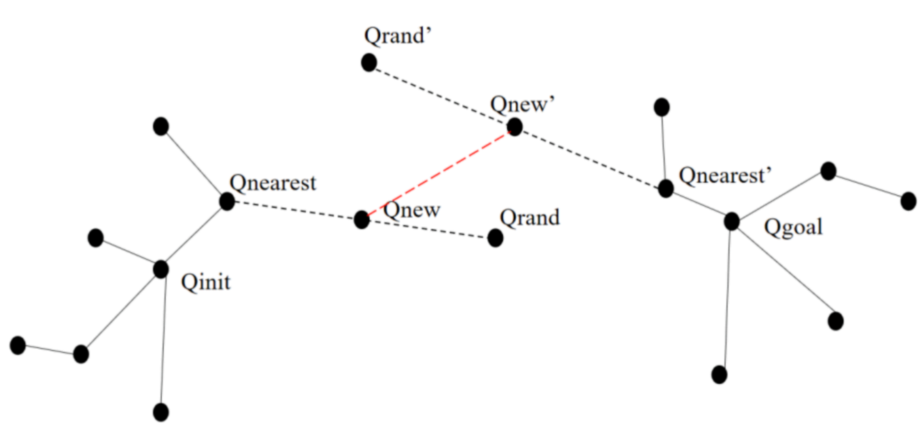


Fig. 9. Schematic diagram of the RRT-Connect algorithm.

Table 7
The pseudo-code of the PRM's algorithm.

Input: *n*,*k*
Result: A road map $G=(V,E)$

$V \leftarrow \emptyset$
$E \leftarrow \emptyset$
while $|V| < n$ **do**
repeat
$q \leftarrow$ a random configuration in Q
untile q is collision-free
$V \leftarrow V \cup \{q\}$
end while
for all $q \in V$ **do**
Nq ← the k closest neighbors of q chosen from V according to dist
for all $q' \in N_q$ **do**
if $(q,q') \notin$ E and $\Delta(q,q') \neq$ NIL then
$E \leftarrow E \cup \{(q,q')\}$
end if
end for
end for

Here *n* stands the number of nodes to put in the roadmap, *k* stands the number of closest neighbors to examine for each configuration.

Table 8
Comparison of the different SBM algorithms.

Method	Optimal Path	Real time	Memory	Safety
RRT	N	N	N	Y
RRT-Connect	N	N	N	Y
RRT*	Y	N	N	Y
PRM	N	N	N	Y

Table 9
The pseudo-code of the Dijkstra's algorithm.

Input: *G*,*s*,*t*
Output: *a path from s to t with the minimum weight*

for *each vertex v* **do**
$\{status[v] = 0; wt[v] = -1; dad[v] = -1;\}$
$status[s] = 2; wt[s] = +\infty$;
for *each edge* $[s,w]$ **do**
$\{status[w] = 1; wt[w] = weight(s,w); dad[w] = s;\}$
while *there are fringes* **do**
$v =$ *the fringe with the* min *wt-value*
$status[v] = 2;$
for *each edge*$[v,w]$ **do**
case 1. $status[w]==0;$
$\{status[w] = 1; wt[w] = wt[v] + weight(v,w); dad[w] = v;\}$
case 2. $(status[w]==1)$ and $(wt[w] > (wt[v] + weight(v,w));$
$\{wt[w] = wt[v] + weight(v,w); dad[w] = v;\}$

Here *G* stands a graph, *s* stands the starting point of mobile robot, *t* stands the target point.

efficiency.

The RRT* algorithm has been shown to be an asymptotically optimal path planning method. In other words, when the number of samples tends to infinity, the solution obtained by RRT* is the optimal solution that converges with the probabilistic solution. Recent research developments have proposed many improved RRT* algorithms that use different methods to improve the performance of RRT*. Qi et al. (2020) extended the RRT* algorithm to an unknown dynamic environment and verified the effectiveness of the new algorithm from both simulation and real experiments. Brunner et al. (2013) used biased sampling to improve RRT* by controlling the direction of the RRT* algorithm iteration by biasing the operation to distribute the samples in our desired region on the one hand, making it more efficient. The RRT* algorithm based on heuristic information-guided local sampling was also proposed (Ryu & Park, 2019), where the principle is to use heuristic information to guide the search to find the optimal path quickly after RRT* finds the feasible path for the first time.

The paths obtained by the above methods are often convoluted, so based on the RRT* algorithm, the RRT*-smart algorithm was proposed in Islam's literature (Islam et al., 2012). RRT*-Smart is identical to RRT* in the first stage of operation, but after finding a feasible path from the starting point to the end point it starts to consider optimizing the path and turning the curve into a straight one. The process actually starts from the starting point and keeps looking for whether it can connect directly to the preceding nodes without any obstacles. If you connect one layer directly forward there is one more straight line and one less curve.

In addition to the above classical optimization schemes, the optimization about RRT series of algorithms has been very popular in recent years. In the Xu's literature, a simplified map-based regional sampling RRT* algorithm (simplified map-based regional sampling RRT*, SMRS-RRT*) was proposed for the shortcomings of RRT* algorithm such as low search efficiency (Xu et al., 2020), slow convergence speed and excessive memory consumption. In the Wang's literature, a path planning method based on an improved RRT* algorithm was proposed for solving the path planning problem of mobile robots based on articulated structures and drifting environmental conditions with vectorized maps and dynamic constraints to achieve robot motion (Wang et al., 2022a). In the Li's literature, an improved motion planning algorithm (fast-RRT*) based on a hybrid sampling strategy and a backtracking-based adaptive GA is proposed (Li et al., 2022a). Researchers have been working on the optimization of the RRT algorithm, focusing on three main aspects: (a) the random sampling approach. (b) The strategy of branch expansion. (c) The combination with mobile robot kinematics. These methods have the advantages of fast search of free space and real-time collision detection. However, this class of methods produces unstable paths with heavy dependence on the nearest node, and additionally requires complete environmental information for node collision detection (Wang, 2020).

PRM (probabilistic roadmap method)

The PRM proposed by Overmars et al. in the early 1990s (Chen et al., 2021a). This method is based on the sampling algorithm, which solves the difficulty of constructing an effective path graph in high-dimensional space. This algorithm represents the connectivity of path graph by sampling in configuration space, collision detection of sampling points, and testing whether adjacent sampling points can be connected (pseudo-code as shown in Table 7).

One of the great advantages of this method is that its complexity mainly depends on the difficulty of finding the path, and has little relationship with the size of the whole planning scene and the dimension of the configuration space. However, when the planned path needs to pass through dense obstacles or narrow passages, the efficiency of PRM becomes low. Researchers have done a lot of research to improve these deficiencies. Zou et al. (2019) introduced an improved strategy of adding nodes on the basis of the basic PRM algorithm to optimize the sampling points trapped in obstacles, but the increase in the number of nodes makes the computing cost higher, which affects the real-time performance of the algorithm; Ravankar et al. (2020) used hierarchical hybrid PRM algorithm and APF method to carry out global path planning, and used a node distribution decomposition method to divide the road map into high potential and low potential areas, which improved the efficiency of the algorithm in searching the path; Inspired by the adjacent correlation matrix, Esposito and Wright (2019) proposed a processing algorithm to optimize the probability roadmap, simplifying the calculation required for processing the number of convex elements and nodes in free space.

To sum up, SBM has stronger search ability when the spatial

Table 10
The pseudo-code of the A*'s algorithm.

```
Mark P[star] as openlist.
while openlist ≠ empty do
  Select the node P[i] from the openlist whose value of evaluation function F(P[i]) is smallest.
  Mark P[i] as closelist.
  if P[i] = P[end] then
    return "path is found'.
  else
    Select the successor node P_i[j] around the node P[i], and calculate F(P_i[j]).
    if P_i[j] belons to obstacle or closelist node then
      continue;
    end if
    Mark P_i[j] as openlist.
    if P_i[j] belons to openlist and F(P_i[j]) < F(P_m[j]) when P[m] was marked as closelist then
    Set parent node of P[j] as P[i],F(P[i]) = F(P_i[j]).
  end if
  end if
end while
return "the path cannot be found".
Mark P[star] as openlist.
while openlist ≠ empty do
```

Here $P[star]$ stands the starting point of mobile robot, $P[end]$ stands the target point.

Table 11
Comparison of the different cell decomposition algorithms.

Method	A*	Dijkstra
Advantage	Global optimum; Introducing heuristic information for higher efficiency	Global optimum
Disadvantage	Lack of adaptability	Low efficiency; Lack of adaptability
Personalized improvement scheme	Optimize heuristic function; Change environment modeling method	Hierarchical search; Improve data structure; Add heuristic information
Common improvement scheme	Two way search, expand the search angle; Introducing path optimization rules	

dimension is high or the size is large. However, SBM is sampled in the whole space, so the amount of calculation is large, and there is still much room for improvement in planning efficiency. For PRM, if the sampling parameters are not set properly, even if there is a reasonable path in the space, the complete path may not be found. In order to visually compare SBM algorithms, we have done so in Table 8.

3.1.3. GSA (graph search algorithm)

(1) Dijkstra algorithm.

Dijkstra's algorithm was proposed by Dutch computer scientist Edsger Wybe Dijkstra in 1959, and it has been successfully applied to 2D path planning for mobile robots, computer science, geographic information science, and transportation (Wei et al., 2019; Alyasin et al., 2019; Yu & Ge, 2017). The iterative process of path planning of traditional Dijkstra algorithm is shown in the Table 9 (Sun, 2017a).

Huang (2018b) introduced the concept of equivalent path and determined the conversion formula between equivalent path and actual path, thus improving the traditional Dijkstra's algorithm. Dijkstra's algorithm is a graph search algorithm for single-source optimal path. It performs node expansion by first ranking the total path cost $F(n)$ of each node, and then selecting the node with the lowest cost to expand until reaching the target location. The ranking property of the algorithm determines that the algorithm must be extended to all nodes whose total cost is less than that node before extending to a certain node, which results in the low planning efficiency of Dijkstra's algorithm, so the method is often used as global path planning for mobile robots or local path planning for scenes with few nodes Sun (2021). So far, there are three main approaches to improve the traditional Dijkstra algorithm. One is to analyze and improve the space complexity of the algorithm to improve storage efficiency and save space. For example, Fink et al. (2019) applied Dijkstra's algorithm to a planetary rover in a 3D environment, Balado et al. (2019) applied Dijkstra's algorithm in combination with point cloud maps for path planning in urban 3D environments, and Yu et al. (2018) introduced Dijkstra's algorithm in the field of path planning for driverless cars. The second one is to analyze and improve the time complexity of the algorithm. The traditional Dijkstra's algorithm is inefficient with a long running time, in order to improve the operational efficiency and reduce the time complexity, Dijkstra's algorithm is improved and applied to dynamic scenarios (Guo et al., 2019; Zhang et al., 2018b). The third one is to apply the algorithm in different fields to develop the application space of the algorithm and enrich the application scope of the algorithm, which has been studied by many scholars. Souza et al. (2019) applied Dijkstra's algorithm to transportation cost calculation in the mining industry, Santos et al. (2019) introduced Dijkstra's algorithm in the field of power transmission, and Tang et al. (2018) introduced Dijkstra's algorithm in broadband technology.

(2) A* algorithm and the optimization algorithm based on it.

The A* algorithm is an improved version of Dijkstra's algorithm, which uses a heuristic function $H(n)$to filter the extended nodes, greatly reducing the number of total extended nodes and improving the efficiency of trajectory planning. The A* algorithm is a heuristic search algorithm for solving the shortest source path in a static environment (Wang et al., 2022c). The principle of the algorithm is shown in Table 10.

In the Grid map environment, the A* algorithm can effectively represent arbitrarily shaped obstacles in two-dimensional space. However, in the three-dimensional environment, the valuation function becomes complicated with the increase of dimensionality, so the A* algorithm is less applied in the high-dimensional space. To overcome this drawback, many studies have proposed optimization schemes based on A* algorithms, such as Dynamic A* (D*), Field D*, Theta*, Anytime repairing A* (ARA*), and so on (Wang, 2020). The D* algorithm was proposed by Stentz (1997), which is based on the principle of first generating an initial path with the help of the A* algorithm, and then relying on the on-board sensors to detect the external environment during the operation of the mobile robot to correct the initial path in real time. The Field D* algorithm was proposed by D. Ferguson and A. Stentz in 2006 (Ferguson & Stentz, 2006), which is based on the D* algorithm and introduces a linear interpolation function as a way to generate smooth paths. Daniel et al. (2010) proposed the Theta* algorithm in 2007, which also aims to smooth the paths generated by the A* algorithm as a way to save the energy consumption of the mobile robot. The ARA* algorithm, on the other hand, automatically adjusts the search results based on the available planning time (Ziegler et al., 2008), reusing the previous search results in the process of search path optimization, and is therefore more efficient than other algorithms. In addition to these classical optimization strategies, researchers have done a lot of work in recent years to improve the A* algorithm. In the literature (He et al., 2022), an obstacle avoidance algorithm for multi-vessel encounters is proposed, which generates a quaternion ship domain based on automatic identification system data and calculates the ship navigation risk cost by combining the quaternion ship domain and the potential field to ensure the safety of the ship. Martins et al. (2022) designed an improved multi-objective A* (IMOA*) algorithm for path planning of mobile robots in large workspaces. Wang et al. (2022b) introduced road condition indicators in the evaluation function of the A* algorithm to reduce the road cost of the mobile robot while walking.

In order to visually compare CD algorithms, we have done so in Table 11.

3.1.4. Artificial potential field algorithm (APF)

Platonov in 1974 presented the APF approach. The method is to

Table 12
The pseudo-code of the APF's algorithm.

$x(0) = x_{\text{start}}$
$y(0) = y_{\text{start}}$
$\theta = \theta_{\text{start}}$
$i = 0$
while $\nabla U(x(i)) \neq 0$ and $\nabla U(y(i)) \neq 0$ **do**
$\theta = \text{arctant}(\nabla U(x(i))/\nabla U(y(i)))$
$x(i+1) = x(i) + \gamma^{*}\cos(\theta)$
$y(i+1) = y(i) + \gamma^{*}\cos(\theta)$
i++
end while

Here x_{start} stands the X-axis coordinate of mobile robot starting point, y_{start} stands the Y-axis coordinate, θ_{start} stands the included angle between the line between the starting point and the coordinate point and the positive direction of the X axis.

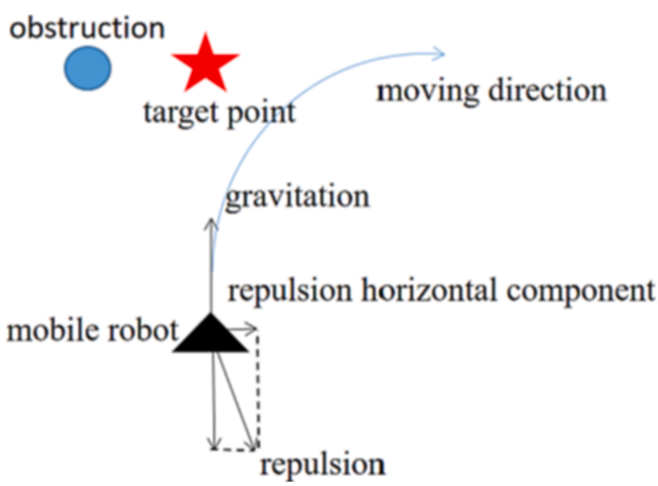


Fig. 11. Inaccessibility of targets.

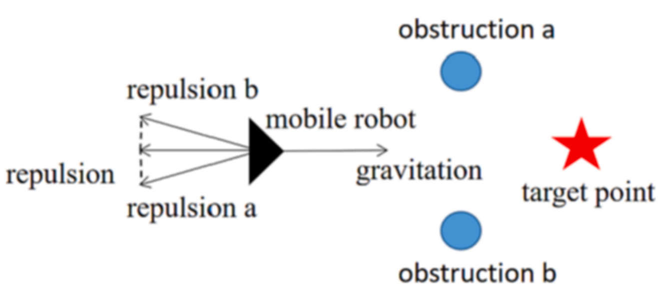


Fig. 12. Optimum local solution.

construct virtual gravitational force and virtual repulsive force in the environment where the robot is located. The virtual gravitational force is generated by the target point and the virtual repulsive force is generated by the obstacle, and the robot moves toward the target point under the joint action of gravitational force and repulsive force (Zhang and Ming, 2021d; Liu et al., 2022). The core of APF is gradient descent method, which is a common method to solve optimization problems. From the initial state, move in the opposite direction to the gradient. The pseudo code of APF is shown in the Table 12.

Through analysis, it can be known that when the mobile robot is very close to the target point, the gravitational force is small. If there is an obstacle nearby at this time, the repulsive force will be large, and the combined force will cause the robot to deviate from the target point, and such a situation is called target unreachable (as shown in Fig. 11). On the other hand, in the process of robot movement, there may be a situation where the combined force and the repulsive force are equal in size and opposite in direction, and the robot falls into a local stable state and cannot continue to move forward, which is called a local optimal solution (as shown in Fig. 12). In addition, due to the mismatch between the step size of the mobile robot and the control distance of the obstacle and the discretization of the solution process, APF is prone to the problem of path oscillation. In order to overcome the shortcomings of the artificial potential field algorithm, Du et al. (2021) proposed an improved APF path planning algorithm for unknown environments and dynamic obstacles, and simulation experiments showed that the improved algorithm could resolve the path planning failure with a success rate of 95%.

The existing literature on the application of APF to mobile robot path planning is mainly focused on solving the APF local minima problem by optimizing and improving the gravitational and repulsive potential functions or adding other additional conditions. Yun and Tan (1997) proposed a simple and practical algorithm for escaping local minima in the potential field based motion planning. The algorithm switches between wo control modes: the potential field based control mode and the wall-following control mode. The new algorithm switches to a wall-following control mode when the robot falls into a local minimum. Szczepanski et al. (2022) proposed top quark-based mechanism which can prevent the robot from stagnation in a local minimum, and allow to bypass the local minimum with a smooth movement.

A MTAPF (multiple sub-target artificial potential field) deterministic algorithm with improved APF algorithm is proposed by Sang et al. (2021), where the global optimal path is first generated by the improved heuristic A* algorithm, and the optimal path is divided into multiple sub-target points to form a sequence of sub-target points. MTAPF can greatly reduce the probability of a mobile robot falling into local minima and help the robot to get out of local minima by switching target points. An optimization framework based on the APF model is proposed by Pang et al. (2021) for changing crossing waypoints, route segments, and spatial topological maps of prohibited, restricted, and hazardous areas. Based on this framework, an adaptive approach is proposed to dynamically control the optimization process with the objective of minimizing the total route cost. Using the APF algorithm, a neural network training scheme is proposed by Wang et al. (2021), the robot's workspace is divided into two parts, Global safety area and local hazard area. In the global safety zone, the robot receives only the gravitational force from the target and moves directly toward it. In the danger region, in addition to attractive forces, the robot receives repulsive forces from obstacles. The repulsive force and the angle between the obstacle and the target are then used as inputs to the fuzzy control system to output the deflection angle of the robot, and the final direction of movement of the robot is determined by summing this deflection angle and the direction of attraction. An improved APF was proposed by Luo et al. (2019b). The repulsive force of obstacles outside a certain angle range in the direction of robot movement is excluded to reduce the computational effort; the distance factor between robot and target point is introduced in the repulsive force function to solve the target unreachability problem; the tangent method is used to solve the problem of local minima formed by a single obstacle, and the search method is used to solve the problem of local minima formed by multiple obstacles acting simultaneously. The adaptive step size adjustment algorithm is proposed considering the complexity of path planning. A MTAPF (multi-sub-target artificial potential field) deterministic algorithm based on improved APF is proposed by Shin and Kim (2021). MTAPF belongs to local path planning algorithm, which refers to a global optimal path generated by an improved heuristic A* algorithm, by which the optimal path is divided into multiple sub-objective points to form a sequence of sub-objective points. Zheyi & Bing (2021) added an angle function to the conventional algorithm to match with the Force Field function to improve the effectiveness of the conventional algorithm. Feng et al. (2021) introduced a safe distance in APF so as to generate a safe path in a simulated traffic scenario. Sepehri & Moghaddam (2021) combined the advantages of the RRT algorithm and APF, first using RRT to generate a feasible path with attractive and repulsive potential fields implemented

Table 13
The pseudo-code of the DWA's algorithm.

1.	**BEGIN DWA**(*robotPose,robotGoal,robotModel*)
2.	*desiredV* = *calculateV*(*robotPose,robotGoal*)
3.	*laserscan* = *readScanner*()
4.	*allowable_v* = *generateWindow*(*robotV*, *robotModel*)
5.	*allowable_w* = *generateWindow*(*robotW*, *robotModel*)
6.	**for** each *v* in *allowable_v*
7.	**for** each *w* in *allowable_w*
8.	*dist* = *find_dist*(*v,w,laserscan,robotModel*)
9.	*breakDist* = *calculateBreakingDistance*(*v*)
10.	**if** (*dist* > *breakDist*) //can stop in time
11.	*heading* = *hDiff*(*robotPose,goalPose*, *v,w*)
12.	*clearance* = (*dist-breakDist*)/(dmax - breakDist)
13.	*cost* = *costFunction*(*heading,clearance*, *abs*(*desired_v* - *v*))
14.	**if** (*cost* > *optimal*)
15.	*best_v* = *v*
16.	*best_w* = *w*
17.	*optimal* = *cost*
18.	set robot trajectory to *best_v*, *best_w*
19.	**END**

Here *robotPose* stands the current position of the robot, *robotGoal* stands the target point, *robotModel* stands the robot model.

Table 14
The pseudo-code of the GA's algorithm for path planning.

INPUT: *map,start point,target point, population,maximum number of iteration*	
OUTPUT: *optimal path*	
1.	*SEE* ← See(*map,target point*) *//Traverse the set of visual points*
2.	*X* ← *Initial*(*map,starting point,target point, population*) *//Initial population*
3.	*fx* ← fitness(*X*) *//Calculate initial fitness*
4.	*gen* ← 1
5.	**while** *gen* < *maximum number of iteration* **do**
6.	*fit* ← 1./(*fx* + *1*) *//Calculate the average fitness.*
7.	*Xsel* ← select(*X,fit*) *//Select operation.*
8.	*Xsel* ← crossover(*Xsel,fit*) *//Crossover operation(Xsel).*
9.	*Xsel* ← mutation(*Xsel,fit,map*) *//Mutation operation.*
10.	*X* ← evolving(*Xsel,SEE,map*) *//Evolving operation.*
11.	*fx* ← *fitness*(*X*) *//Calculate the new fitness*
12.	*gen* ← *gen* + 1
13.	**end while**
14.	**return** *optimal path*

Here *map* stands the environment map, *start point* stands the starting point of the mobile robot, *target point* stands the target point, *population* stands initial population and *maximum number of iteration* stands iterations.

at all points along the path and used in a gradient optimization algorithm to determine joint trajectories to reach the desired position. In the literature by Liu et al. (2021d), the two potential fields fused adaptive path planning system consisting of two parts, a potential fields fusion controller and an adaptive weight assignment unit is proposed.

In summary, the Artificial Potential Field Method (APF) has its unique advantages in local path planning for mobile robots, which does not need to generate a complete trajectory in advance, but only determines the next stage of motion based on the current state and the surrounding environment, and this algorithm plans smooth trajectories with less computational effort and can update real-time data to adapt to the environment (Zhang, 2021b).

3.1.5. DWA (dynamic window approach)

DWA is an algorithm proposed by Fox et al. (2002) that combines robot curvature and speed. This algorithm will simulate the trajectory of the robot in a period of time according to the speed, and then it will score these simulated trajectories through scoring rules, transforming the obstacle avoidance problem of the robot into the optimal speed execution problem, which has obvious advantages in feasibility and motion continuity. DWA pseudo code is shown in Table 13.

In DWA, the evaluation function consists of three weighting items: the guide angle of the target point, the distance from the obstacle and the moving speed of the robot. Compared with other obstacle avoidance algorithms, DWA pays more attention to the limitation of the maximum speed and acceleration of the robot. Therefore, from this point of view, the control command of DWA seems to be more in line with the movement of the mobile robot in the actual situation. However, the distribution of obstacles in the robot environment is usually complex, and the weight parameters in the objective function of DWA need to be adjusted to improve the flexibility of the algorithm in different environments. The traditional DWA and its optimization algorithm usually use constant weight parameters when facing different obstacle distributions. In the past few years, many optimized DWAs have also been proposed. Zhang et al. (2015) proposed the combination of DWA and fuzzy logic to analyze the information about targets and obstacles, and use fuzzy logic to adjust the weight combination. The proposed method can make the robot move more smoothly and safely. On this basis, Huang (2018a) proposed a parameter fuzzy control DWA based on radar information to improve the adaptability of mobile robots to complex environments in local path planning. Li et al. (2022b) proposed an improved A* and DWA algorithm, using laser radar to build maps, and realized path planning and automatic navigation of orchard robots. Wu et al. (2022) proposed a hybrid dynamic path planning algorithm based on improved A* and improved DWA. The improved DWA evaluation function can ensure that the local path of the mobile robot is closer to the global optimal path. Zhenyang & Wei (2022) combined A* algorithm with adaptive DWA, and used optimization A* to guide the global path planning of adaptive DWA. This algorithm is more consistent with the motion characteristics of mobile robots.

3.2. Bionic algorithm

Bionic algorithm is a random algorithm inspired by the phenomenon of biological swarm intelligence in nature. Based on the principle and path search mechanism of bionic algorithm, it can be divided into functional bionic algorithm, informational bionic algorithm and component bionic algorithm. Functional bionic algorithm is an algorithm proposed to simulate the role of a specific physiological function of an organism in foraging, social living, escape or other biological behaviors. For example, genetic algorithm, cuckoo search algorithm, sparrow search algorithm. Information bionic algorithm is an algorithm proposed to simulate the basic principles of information transmission and intelligent behavior of organisms in organs, individuals and groups. For example, Particle swarm optimization, bacterial foraging optimization, artificial bee colony algorithm, grey wolf optimizer. component bionic algorithm simulates the behavior and function of organisms from the perspective of chemistry, and realizes the activity behaviors such as mate seeking and aggregation by simulating the secretion and release of trace chemicals of insects, such as ant colony optimization algorithm and firefly algorithm (FA) (Li et al., 2021).

3.2.1. GA (genetic algorithm)

GA was formally proposed by Holland and his students in 1975 , which and applied to optimization, machine learning and other issues. Its core is to use genetic operators to simulate selection, crossover and variation in natural evolution and produce new populations that are more adaptable to the environment (pseudo-code as shown in Table 14) (Xu et al., 2021; Lin et al., 2021; Mahmud et al., 2019).

Lamini et al. (2018) proposed an improved crossover operator, which is used to solve the path planning by GA in static environment. The proposed crossover operator avoids premature convergence and provides a better and more feasible path than its parent generation.

Table 15
The pseudo-code of the ACO's algorithm for path planning.

```
procedure ACO
for each edge
set τ0.
end for
while not stop
for each ant k
randomly choose an initial city.
for i = 1 to n
choose next city j with the probability
given
end for
end for
compute the length Ck of the tour constructed by the kth ant.
for each edge
update the pheromone value
end for
end while
end procedure
```

Here τ_0 stands the initial pheromone value, n stands the number of ants.

Therefore, the convergence rate of this algorithm is faster. Nazarahari et al. (2019) proposed an improved GA, which can improve the original path in continuous space and find an optimal path between the original position and the destination position, as well as extended to the path planning of multiple mobile robots. Shivgan & Dong (2020) proposed a path planning model of UAV based on the acceleration, deceleration, hovering and turning of UAV, considering the total energy consumption of mission. It can reduce the energy consumption of UAV by reducing the number of turns of UAV. Rahmaniar & Rakhmania (2022) proposed an optimal path planning method based on GA. Firstly, find the feasible path points by performing local search. Then, optimize these points to find shortest path. When calculating the optimal path, the position of the points on the path will be smoothly move to avoid obstacles in the environment, which makes this method have advantages in dynamic environment. Chen et al., (2021c) developed a new fitness function of GA, which can consider the constraints, such as obstacles of freedom, length of the path, smoothness of the path, and visibility in camera roaming process. In addition, a new evolution operator is introduced in GA, greatly reducing the iteration times and improving the efficiency of GA. In order to improve the ability of receiving GPS (global positioning system) signals for wall-inspecting robot, Tao et al. (2022) introduced signal influencing factors into the fitness function of GA. Germi et al. (2018) fused GA with APF and realized the adaptability of the algorithm by changing the population generation method of GA in each iteration. Liu et al. (2021b) designed an AGA (adaptive GA) to solve the task allocation, and a fitness function considering the current iteration times and the maximum iteration times is proposed in order to improve the convergence performance of AGA.

3.2.2. ACO (ant colony optimization)

In 1992, Dorigo et al. (1991) proposed ACO in his doctoral dissertations, which was inspired by the behavior of ants discovering paths when they search for food (Dai et al., 2019; Chu et al., 2021)(pseudo-code as shown in Table 15).

ACO is featured by positive feedback, parallel computation, and good robustness performance, but suffers from such disadvantages as slow convergence, easy to fall into local optimum, and premature convergence (Zhang et al., 2020b). In order to address the problems of basic ACO, many scholars have proposed many effective improvement measures in terms of algorithmic framework and structure, such as ACS (ant colony system) (Dorigo & Gambardella, 1997), MMAS (max–min ant system) (Stützle & Hoos, 2000), AS_{elite} (ant system with elitist strategy), AS_{rank} (ant system with elitist and ranking) (Bullnheimer & Hartl, 1997), and other classical improvement algorithms. In spite of effective improvement of optimization capability, these improved algorithms adopt a fixed pattern to update the pheromone and probability transition rules, which leads to the lack of flexibility and the failure of solving the premature convergence of the algorithm.

For the mobile robot path planning problem, many scholars proposed new improvement methods based on existing improvement strategies. Jiao et al. (2018) proposed a polymorphic ACO for path planning, adaptive state transfer strategy and adaptive pheromone update strategy was applied to ensure the relative importance of pheromone intensity and heuristic information in the iteration of algorithm. Miao et al. (2021) proposed an IAACO (improvement adaptive ant colony algorithm). Firstly, in order to speed up the real-time and safety of robot path planning in IAACO, angle guiding factors and obstacle exclusion factors were introduced in the transition probability of ACO; secondly, adaptive adjustment factors and adaptive pheromone volatility factors were introduced in the pheromone update rule of ACO to balance the convergence and global search capability of ACO; finally, the multi-objective performance index was introduced in ACO, which transformed the path planning problem into a multi-objective optimization problem, realizing the global integrated optimization of robot path planning. Ji & Liu (2022) used ACO to find a fastest path for the robot, traversing all the target points. Yang et al. (2022) improved the traditional ACO in terms of both pheromone updates and heuristic functions, providing a strategy for solving the deadlock problem. Combined with other methods, ACO is featured by strong robustness, fast optimization search, and strong convergence. Although these fusion strategies can learn from others to effectively improve the ACO's optimization-seeking ability, the algorithm become more complex and require more optimization time. Chen et al. (2021b) proposed an improved ACO-APF algorithm based on grid maps, which is used in local and global path planning of unmanned boats in dynamic environments. On the basis of ACO, a single-step search strategy was placed by a multi-step search strategy, the pheromone update mechanism was redesigned by Xue et al. (2021), and path smoothing was disposed to improve the performance of the algorithm. Yan (2021) improved the initial pheromone, transition probability and pheromone update strategy. An adaptive ACO path planning method was proposed by Li et al. (2021). The algorithm optimized the initial pheromone matrix based on the environment map, which reduced the blindness of the initial ant colony in path finding and improved the convergence speed. Then an adaptive heuristic function was used to adjust adaptively according to its different proportions in the process of the algorithm, preventing the algorithm from falling into local optimum. The pheromone was updated according to the corner points of the planned route to reduce the sharp angles and unnecessary turns of the route and further optimize the route. Zhu et al. (2021) proposed an algorithm that can adaptively adjust the pheromone coefficients, which can allocate pheromone with more suitable proportion. Zhang et al. (2021c) used the non-uniform distribution to construct the initial pheromone, reducing the blindness of the search in the initial stage, as well as a diffusion model was used to enhance the exploration and collaboration among ants. Wang et al. (2021c) proposed the heuristic information by the directional information of the starting, ending points and the turning angle. The improvement of the heuristic information increased the search direction and reduce the turning angle of the robot. Meanwhile, the MMAS was used to limit the pheromone concentration, preventing from falling into the local optimal path. Chen & Zhou (2021) proposed an improved ACO to search and select the path autonomously according to the number of pheromones, which will be improved by adjusting the heuristic factor function. An efficient gain-based dynamic green ant colony optimization was proposed by Sangeetha et al. (2021a). When path planning is performed in dynamic scenarios, the energy consumption will be significant due to its characteristics. The proposed algorithm reduced the total energy consumed in the path planning process through an efficient pheromone enhancement mechanism based on gain function.

3.2.3. PSO (particle swarm optimization)

PSO was firstly proposed by Eberhart & Kennedy (Jain et al., 2022),

Table 16
The pseudo-code of the PSO's algorithm for path planning.

for each particle i
Initialize velocity Vi and position Xi for particle i
Evaluate particle i and set $pBesti = Xi$
end for
$gBest = \min\{pBesti\}$
while not stop
for $i = 1$ to N
Update the velocity and position of particle i
Evaluate particle i
if fit(Xi) < fit($pBesti$)
$pBesti = Xi$;
if fit($pBesti$) < fit($gBest$)
$gBest = pBesti$;
end for
end while

Here Vi and Xi stand the velocity and position of particle i, N stands the number of ants.

Table 17
The pseudo-code of the BFO's algorithm for path planning.

Initialize the parameters
Initialize a random population $\{x_i\}$ for $i \in [1,N]$
For $l = 1$ to N_e
For $k = 1$ to N_r
For $j = 1$ to N_c
For each individual $x_i, i \in [1,N]$
Generate a random n-dimensional unit vector Δ
For $m = 1$ to N_s(cost reduction steps)
$\hat{x}i \leftarrow xi + c\Delta$
If $f'(\hat{x}i) < f'(xi)$ **then**
$xi \leftarrow \hat{x}i$
else
$m \leftarrow Ns$(exit the cost reduction loop)
End if
Next m
Next individual
Next j
For each individual $x_i, i \in [1,N]$
$F_i \leftarrow$ average value of $f'(xi)$ during N_c steps of chemotaxis loop
Next individual
Eliminate the worst $N/2$ individuals based on $\{F_i\}$
Clone the best $N/2$ individuals based on $\{F_i\}$
Next k
For each individual $x_i, i \in [1,N]$
Random number $r \leftarrow U[0,1]$
If $r < p_e$ **then**
$x_i \leftarrow$ random point in the search space
End if
Next individual
Next l

Here N stands the number of iterations, N_e stands the number of dispels, N_r stands the number of copies, N_c stands the number of chemotaxis.

and originated from the study of foraging behavior of birds(pseudo-code as shown in Table 16).

Particle swarm algorithm is a representative algorithm that combine particle swarm algorithm and other algorithms or factors. The final optimization goals of hybrid particle swarm algorithm are: (a) to improve the convergence and diversity of the population, and prevent premature convergence; (b) to obtain the optimal solution quickly and reduce the time cost; (c) to improve the local search performance and accuracy of PSO. At present, there are usually two improvement directions to achieve the above objectives (Yang & Cai, 2021).

(1) Adjust the parameters of the PSO to balance the global detection and local exploitation capabilities. For example, Chai et al. (2021) applied a locally improved multi-objective particle swarm optimization method to deal with the problem about constrained ballistic path planning. Zhang et al. (2020a) improved the inertia weight, acceleration factor, and localization to prevent the algorithm from falling into local minimum and improve the convergence speed of the algorithm. Then, use fitness variance to measure the diversity of the particles. The increase in diversity helps to overcome the prematureness. The diversity of the particles is increased by the defined extended Gaussian distribution. Finally, apply the smoothing principle to path planning. Phung & Ha (2021) proposed a new spherical vector-based particle swarm optimization.

(2) Combining PSO and other optimization algorithms can form a hybrid PSO algorithm. Qiuyun et al. (2021) proposed an improved particle swarm optimization algorithm, which can better solve the problem of path planning. We propose a new path planning method, and design a crossover operation in GA based on the encoding method of this algorithm to update the locations of particles. In addition, a variational mechanism is used in order to prevent the algorithm from local optimum. Wu et al. (2021b) proposed an effective compensation method based on reinforcement learning and particle swarm optimization, which provided a strategy for real-time rescue and assignment of multi-AUV systems in a 3D underwater environment.

3.2.4. BFO (bacterial foraging optimization)

In 2002, Passino proposed BFO based on the simulation of foraging and reproduction behavior of bacterial population (Yan et al., 2018; Passino, 2002). Beginning from initializing a set of random solutions, regarding the locations of bacteria as potential solutions to the problem, the algorithm change the locations of bacteria by simulating the chemotaxis, reproduction, and elimination-dispersal behaviors of the bacterial population to optimize them(pseudo-code as shown in Table 17).

Due to its simplicity and efficiency, BFO has been widely used in the path planning of mobile robots. Muni et al. (2021) used the foraging quality of the Escherichia coli to obtain the shortest path between two locations in the least amount of time. The Gaussian cost function assigned to the bacterial attractor and repellant profiles plays a major role in obtaining the best path between two locations. Long et al. (2020) proposed a grid partition-based BFO algorithm. And the chemotactic operation that using A* optimize this algorithm allow this algorithm to be applied in unmanned surface mobile robots. Abdi et al. (2020) generated virtual obstacles at local minima to make the robot shrink and regenerate a safe path. This method can be easily extended to coordinated movements between multiple robots. The information associated with the virtual obstacle is shared with the whole population, which allowed them to escape the same local minima, thus saving time and effort. Long et al. (2021) proposed a new multi-subpopulation bacterial foraging optimisation algorithm for path planning of sleep robots to improve the search performance of mobile robots in complex environments. From the above-mentioned optimized algorithm improvements and applications of bacterial community, there have been significant advances in the research of bacterial optimization since the past decades or so. Parameter improvement, multi-algorithm hybridization, and operator improvement will remain important directions for algorithm improvement research. However, there are still some deficiencies in the existing research, which deserve to be further studied. Wang et al. (2022d) proposed an IBFOA (improved bacterial foraging optimization algorithm). In addition, in order to overcome the problems in optimization algorithms and traditional interpolation algorithms, such as large amount of calculation, unsmooth trajectory, etc., adaptive factors and elite retention strategies were used to implement IBFOA. Wang et al. (2021d) designed an adaptive decreasing fractal dimension chemical step instead of fixed step to achieve adaptive step length adjustment, proposed an optimal swimming search method to solve the problems of traditional BFOA, such as invalid search and repeated search. In addition, an adaptive migration probability was developed to replace the fixed migration probability, which can prevent elite individuals from being lost in BOF.

3.2.5. ABC (artificial bee colony algorithm)

ABC is an algorithm inspired by bee colony behavior, proposed by

Table 18
The pseudo-code of the ABC's algorithm for path planning.

Input: *N,FN,lb,ub,limit*
Output:
bestSolutionFound
INITIALIZATION PHASE:
for $m = 1$ to *FN* **do**
Random selection of food source x_m
end for
for *iteration* $= 1$ to *N* **do**
EMPLOYED BEES PHASE:
for $m = 1$ to *FN* **do**
Select r_m randomly
Produce new solution v_m
if fitness(v_m) > fitness(x_m) **then**
Set x_m to v_m
Set $trials_m$ to 0
else
Increment $trials_m$
end if
end for
ONLOOKER BEES PHASE:
for $m = 1$ to *FN* **do**
Calculate selection probability p_m
end for
for $m = 1$ to *FN* **do**
Select r_m
Produce new solution v_m
if fitness(v_m) > fitness(x_m) **then**
Set x_m to v_m
Set $trials_m$ to 0
else
Increment $trials_m$
end if
end for
SCOUT PHASE:
if *iteration* modulus $SPP = 0$ **then**
Select food source with the highest *trials* value
if $trials_m \geq limit$ **then**
Set x_m to random solution in range ($lb \div ub$)
end if
end if
SAVE RESULT:
Update *bestSolutionFound* with position of food source with the best fitness
end for

Here *N* stands the number of iterations, *FN* stands the number of honey sources, *lb*,*ub* stand the lower and upper limit of search space and *limit* stands the same honey source is limited to mining times.

Table 19
The pseudo-code of the CSA's algorithm for path planning.

begin
Objective function $f(x)$, $x=(x_1,\ldots,x_d)^T$
Generate initial population of
n host nests $x_i (i = 1,2,\ldots,n)$
while ($t < MaxGeneration$) *or* (*stop criterion*)
Get a cuckoo randomly by Levy flights
evaluate its quality/fitness F_i
Choose a nest among n (*say, j*) *randomly*
if ($F_i > F_j$),
replace j by the new solution;
end
A fraction (p_a) *of worse nests are abandoned and new ones are built*;
Keep the best solutions (*or nests with quality solutions*);
Rank the solutions and find the current best
end while
Postprocess results and visualization
end

Here $f(x)$ stands the objective function, n stands the number of host nests, *MaxGeneration* stands the number of iterations.

Karaboga's group for optimizing algebra problems in 2005 (Karaboga & Basturk, 2007). The ABC algorithm divides bees into three categories: employed bees, onlooker bees, and scout. The employed bees search for nectar sources, the onlooker bees wait in the hive and select nectar sources, and the scout randomly search for other nectar sources. The location of the nectar source corresponds to the solution of the problem, and the amount of nectar is related to the fitness of the solution.(pseudo-code as shown in Table 18).

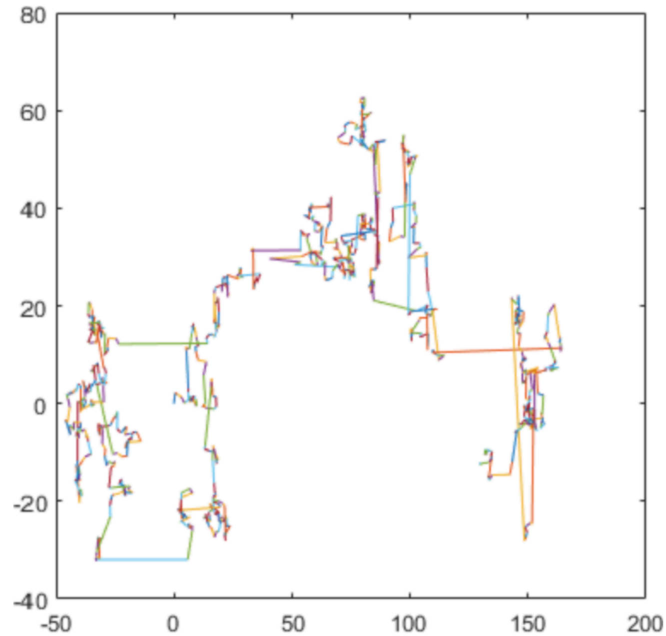


Fig. 13. Levy flight path.

Although ABC is not mature yet and its research is still in the exploration stage, it has been widely concerned by experts and scholars due to its powerful global search capability. In recent years, a large number of researchers have improved the performance of ABC algorithm in three aspects: improvement of the position update equation, the dimensional update strategy, and the overall search strategy (Li, 2020). From the perspective of improving the location update equation, Shah et al. (2018) introduced the gbest mechanism applied in the particle swarm algorithm into the traditional ABC iterative formulation, accelerating the convergence speed. Muntasha et al. (2021) proposed that an anti-collision and path planning system for swarm UAVs can be designed by ABC. The proposed system uses ABC algorithm to optimize the speed of UAVs that can reach the destination efficiently in the shortest path without collisions with others. Chengli et al. (2018) added variable neighborhood search factors to search equation to increase the population diversity and proposed a new empirical strategy to further improve the global search efficiency. From the perspective of improving the dimensional update strategy, Chu et al. (2016) changed ABC's single-dimensional updating strategy to full dimensional, and generated two solutions each time, and chose the better one to improve the search efficiency. From the perspective of improving the overall search strategy, Szczepanski & Tarczewski (2021) proposed a hybrid approach for path planning of variable-space mobile robots. It consists of an offline global path optimization algorithm, in which ABC was applied, and an online path planning scheme. In the path planning layer, Zhao et al. (2021) developed MABCM (multiple-sub populations ABC with memory) algorithm and proposed a new exit evaluation strategy. Hao et al. (2021) proposed the ABC based on balanced search factor for UAV path planning. It used the search strategy based on balanced search factor to perform a deep search while maintaining population diversity. Liu et al. (2021) calculated the optimal path planning between adjacent mission points through improved ABC.

3.2.6. CSA (cuckoo search algorithm)

Yang & Deb (2009) proposed CSA which is an emerging heuristic

Table 20
The pseudo-code of the FA's algorithm for path planning.

1.	Objective function $f(x)$, $x=(x_1,\dots,x_d)^T$
2.	Generate initial population of firefies $x_i (i=1,2,\dots,n)$
3.	Light intensity I_i at x_i is determined by $f(x_i)$
4.	Input γ
5.	**while** ($t < MaxGeneration$)
6.	**for** $i=1{:}n$ all n fireflies
7.	**for** $j=1{:}i$ all n fireflies
8.	**if** ($I_j > I_i$), move firefly i towards j in d-dimension;
9.	**end if**
10.	Attractiveness varies with distance r via exp[$-\gamma r$]
11.	Evaluate new solutions and update light intensity
12.	**end for** j
13.	**end for** i
14.	Rank the fireflies and find the current best
15.	**end while**

Table 21
The pseudo-code of the GWO's algorithm for path planning.

1.	**Begin**
2.	Initialize the grey wolf population $X_i (i=1,2,\dots,n)$
3.	Initialize a,A,and C
4.	**For** all X_i **do**
5.	Calculate fitness $F(X_i)$
6.	**End for**
7.	Get the first three best wolves as X_α,X_β,and X_δ
8.	**While**($t <$ *Max number of iterations*)
9.	**For** each search agent
10.	Update the position of the current search agent
11.	**End for**
12.	Update a,A,and C
13.	**For** all X_i **do**
14.	Calculate fitness $F(X_i)$
15.	**End for**
16.	Update X_α,X_β,and X_δ
17.	$t=t+1$
18.	**End while**
19.	Return X_α
20.	**End**

Here n stands the number of population, a decreases linearly from 2 to 0, A and C stand coefficient vector.

algorithm in 2009. It mainly originated from the brood parasitism reproduction of cuckoo and Levy flight search mechanism (pseudo-code as shown in Table 19 and Fig. 13) (Wu, 2021).

Levy flight is a kind of wandering foraging method in the animal realm. The wandering step length satisfies a stable distribution of heavy tails. And during the wandering process, short-distance exploration is interspersed with occasional long-distance walking, which is a most ideal way of foraging. Using Levy search in intelligent algorithms can prevent from easily falling into local optimal solutions (Liu, 2019). Therefore, CSA has many advantages, such as avoiding local optimum, few and simple parameters, and fast convergence, etc. Once proposed, it has gained the common attention of domestic and foreign scientific scholars. And through continuous improvement and optimization, it has become an active bionic intelligent optimization algorithm nowadays. Wu (2021) proposed a fusion obstacle avoidance algorithm of cuckoo and pigeon flock. Firstly, we use the Levy flight mechanism in CSA to perform long-short alternate search to obtain the latest point of nest location, and then obtain the suboptimal obstacle avoidance path; secondly, we introduce the map-compass operator and the ground surface operator of the pigeon flock optimization algorithm on the basis of the suboptimal path to perform the secondary obstacle avoidance path planning demonstration, update the latest point of nest position again, and then obtain the optimal obstacle avoidance path. Zhang et al. (2021a) proposed an adaptive cuckoo search algorithm, which has high efficiency and good stability under strict dynamic constraints. Sharma et al. (2021) changed the random function in the algorithm based on the traditional cuckoo algorithm, obtaining more alternative solutions and

Table 22
The pseudo-code of the SSA's algorithm for path planning.

Input: G,PD,SD,R_2,n	
Initialize a population of n sparrows and define its relevant parameters.	
1.	**Output:**X_{best},f_g.
2.	**while**($t < G$)
3.	Rank the fitness values and find the current worst individual.
4.	$R_2 = rand(1)$
5.	**for** $i=1{:}PD$
6.	Using equation update the sparrow's location;
7.	**end for**
8.	**for** $i=(PD+1){:}n$
9.	Using equation update the sparrow's location;
10.	**end for**
11.	**for** $i=1{:}SD$
12.	Using equation update the sparrow's location;
13.	**end for**
14.	Get the current new location;
15.	If the new location is better than before,update it;
16.	$t=t+1$
	end while
17.	**return** X_{best},f_g.

Here G stands the maximum iterations, PD stands the number of producers, SD stands the number of sparrows who perceive the danger, R_2 stands the alarm value and n stands the number of sparrows.

avoiding the defect of premature in traditional algorithm. Song et al. (2020) proposed an improved CSA based on compact parallel technology for three-dimensional path planning. Fan et al. (2021) established the collision avoidance model of unmanned craft and obstacles based on the principle of speed obstacle method, and proposed a control method for dynamic collision on the basis of improved CSA.

3.2.7. FA (firefly algorithm)

In 2009, professor Yang (2010) of the University of Cambridge proposed FA by simulating the natural phenomenon of firefly swarming at night (pseudo-code as shown in Table 20).

Here f(x) stands the objective function, γ stands the light absorption coefficient, *MaxGeneration* stands the number of iterations.

Although FA has the same advantages as intelligent algorithms, currently it has an imperfect theoretical foundation, weak interpretability, and suffers from common defects of other swarm intelligence stochastic optimization algorithms. For this reason, many improvements have been made to FA in the academic world. These improvement methods can be summarized into the following categories (Song, 2021): the first is elitism that prevents fireflies form moving to worse positions with poor fluorescein (Wang et al., 2017a; Wang et al., 2016); the second is, random moving mode that uses the random moving part of the algorithm to avoid local optima and enhance the global search of the algorithm (Dhal et al., 2016); the third is adaptive parameter method that change the parameters in the standard FA so that they can adapt to the algorithmic search process (Yu et al., 2016; Wang et al., 2017b; Chou & Ngo, 2017); and the fourth is mixing other algorithms for improvement. As FA has good global search capability but weak local search capability, combining with other intelligent algorithms can make up for its shortcomings (Pei et al., 2019). Improvements in the field of mobile robot path planning are as follows: Patle et al. (2018) applied FA in mobile robot path planning under uncertain situations. Sadhu et al. (2018) proposed a strategy of Q-learning FA was for light absorption parameters. Each firefly learned through a Q-learning strategy and then applied what they learned into the execution phase. An optimal FA-based path planning method with adaptive population size was proposed by Li et al. (2020). Firstly, an evaluation method for collision degree was established at the cost of collision avoidance. Based on the collision degree of

Table 23
Comparison of bionics-based path planning algorithms.

Method	Parallel search capability	Global optimization capability	Convergence time	Whether the search direction can be adjusted	Escape ability of local optimal solution	Robustness	Scope of application
GA	Good	Good	Long	Yes	Weak	Good	Global path planning 2. TSP
ACO	Good	Good	Long	Yes	Weak	Good	Global path planning 2. TSP
PSO	Good	Good	Short	Yes	Weak	Medium	Global path planning Local path planning
BFO	Good	Good	Long	Yes	Good	Medium	Global path planning Multi-robot combinatorial optimization
ABC	Good	Good	Medium	Yes	Good	Good	Global path planning Multi-robot combinatorial optimization Path continuous optimization
CSA	Good	Good	Medium	Yes	Weak	Medium	Global path planning Path continuous optimization
FA	Weak	Good	Long	Yes	Good	Medium	Local path planning TSP
GWO	Weak	Good	Long	Yes	Weak	Medium	Global path planning Multi-robot combinatorial optimization
SSA	Good	Weak	Short	Yes	Weak	Good	Global path planning Multi-robot combinatorial optimization

Table 24
Comparison of bionics-based path planning algorithms.

	BNN	FL
Identical points	It has strong fault tolerance and can process inaccurate information in an imprecise way. When dealing with and solving problems, the precise mathematical model of the control object is not required.	
Differences	It can directly handle structured knowledge. It has poor ability to adapt to environmental changes and acquire knowledge.	It can not deal with structured knowledge directly, and needs to learn samples. It has learning ability and easy knowledge acquisition.
Advantages of combination of the two	Knowledge can be acquired through learning. Strong self-organization and adaptability. Strong robustness.	

the population, two nonlinear functions were proposed to determine the population size. Then, individuals were added or deleted from the firefly population. Individuals can add randomly. The feasible and infeasible solutions were distinguished in the firefly population, and the fireflies in the infeasible solutions were first deleted when performing the elimination operation. Finally, based on the existing methods for handling infeasible paths, a coefficient adaptively adjusted according to the population size was introduced, which can control the degree of infeasible paths approximating the feasible region. A new hybrid algorithm based on GA and FA was proposed by Zhang et al. (2022a). The core idea of this new algorithm is that when FA falls into a local optimal solution, the local optimal fireflies are regarded as a population. Selection, crossover and mutation of population perform in GA, and the optimal individual fireflies can be obtained by genetic operations.

3.2.8. GWO (grey wolf optimizer)

The GWO proposed by Australian scholars Mirjalili et al. (2014) is an intelligent optimization search method. Grey wolves are the canine carnivore animals with high environmental adaptability, which are good at long-distance running at high speed. GWO encompasses their stratification of social hierarchy, the process of besieging and attacking prey(Lu, 2017)(pseudo-code as shown in Table 21).

From the current research on GWO, GWO has been continuously improved in theoretical research and has a broad application prospect in the field of mobile robot path planning. Although the GWO algorithm has many application advantages, it still has some disadvantages (Zhang & Wang, 2019; Tang, 2019): (1) reducing population diversity due to randomly generated initialized populations; (2) slow convergence at a later stage caused by the search mechanism of the GWO algorithm; and (3) tend to fall into local optimum. Zhang et al. (2016) proposed a new *meta*-heuristic GWO for two-dimensional path planning of the unmanned combat air vehicle that can find a safe path by connecting selected nodes in a two-dimensional coordinate system while keeping away from threat areas and saving fuel. Albina & Lee (2019) proposed a multi-robot exploration stochastic optimization method that simulates the collaborative predatory behavior of grey wolves. The motion of the robots is computed by a combination of deterministic and *meta*-heuristic techniques, using a new method of collaborative multi-robot search and GWO as a mixed stochastic search. Jamshidi et al. (2021) proposed an improved GWO for UAV 3D path planning considering UAV dynamics, in which a variable weight called "alignment coefficient" was defined to handle the scattering of waypoint. The application of parallel GWO reduces the computation time and makes real-time implementation more possible. Zhang et al. (2021e) proposed an adaptive convergence factor adjustment strategy and adaptive weighting factors to update individual positions by convergence analysis and test functions. Jarray et al. (2022) proposed a new parallel cooperative coevolutionary grey wolf optimizer to improve the accuracy of UAV routes. Liu et al. (2021) endowed α wolf, β wolf, and δ wolf active search ability and added dynamic weight in the grey wolf position update to prevent wolves from losing diversity and falling into local optimum. Liu & Li (2022) proposed a mixed path planning method combining modified gray wolf optimization with a situation assessment mechanism. Kiani et al. (2021) proposed two UAV 3D path planning methods based on incremental GWO and Extended-GWO to determine the UAVs optimal path with minimum cost and minimum execution time. Fei et al. (2021) used a chaotic random method to generate the initial population, introduced a velocity sorting system into the traditional particle swarm algorithm to propose mixed algorithm of GWO-PSO. Panwar & Deep (2021) proposed the discrete

Table 25
Analysis of various path planning techniques.

Authors	Path planning of		Dimension	Basic Algorithm	Algorithm		Path smooth	Test of		Environmental modeling method
	Global	Local			Optimization	Fusion		Simulation	Actual	
(Abdi et al., 2020)	Y	N	2	BFO	Y	N	N	Y	N	GM
(Albina & Lee, 2019)	N	Y	2	GWO	Y	N	N	Y	N	GM
(AlKhlidi et al., 2021)	Y	N	2	FL/PSO	Y	Y	N	Y	N	–
(Alyasin et al., 2019)	Y	N	2	Dijkstra	N	N	N	N	Y	–
(Balado et al., 2019)	Y	Y	3	Dijkstra	Y	N	N	Y	Y	–
(Bao et al., 2022)	Y	Y	2	APF/A*	Y	Y	N	Y	Y	GM
(Brunner et al., 2013)	Y	N	3	RRT	Y	N	N	Y	N	GM
(Chai et al., 2021)	Y	Y	2	PSO	Y	N	N	Y	N	–
(Challita et al., 2018)	N	Y	2	CD	Y	N	N	Y	N	GM
(Chen et al., 2021a)	Y	N	3	PRM	Y	N	N	Y	Y	GCM
(Chen et al., 2021b)	Y	Y	2	ACO/APF	Y	Y	N	Y	Y	GM
Chen & Zhou (2021)	Y	N	2	ACO	Y	N	N	N	N	–
(Chen et al., 2021c)	Y	N	2	GA	Y	N	Y	Y	N	GM
(Chen & Zou, 2021)	N	Y	2	BNN	Y	N	N	Y	N	GM
(Chengli et al., 2018)	Y	Y	–	ABC	Y	N	N	Y	N	–
(Cheng et al., 2021)	Y	N	2	GWO/PSO	Y	Y	N	Y	N	GM
(Dai et al., 2019)	Y	N	2	ACO/A*	Y	Y	N	Y	N	GM
(Daniel et al., 2010)	Y	N	2	A*	Y	N	Y	Y	N	GM
(Ding et al., 2016)	Y	Y	2	APF	Y	N	Y	Y	N	GM
(Du et al., 2021)	N	Y	2	APF	Y	N	N	Y	N	–
(Esposito & Wright, 2019)	Y	N	2	PRM	Y	N	N	Y	N	GCM
(Fang et al., 2021)	N	Y	2	CSA	Y	N	N	Y	N	–
(Fei et al., 2021)	Y	N	2	GWO/PSO	Y	Y	N	Y	N	–
(Feng et al., 2021)	N	Y	2	APF/FL	Y	Y	N	Y	N	–
(Ferguson & Stentz, 2006)	Y	Y	2	A*	Y	N	Y	Y	Y	GM
(Fink et al., 2019)	Y	N	3	Dijkstra	Y	N	Y	Y	N	–
(Fox et al., 2002)	Y	Y	3	DWA	N	N	N	Y	N	GCM
(Gao et al., 2021)	Y	N	2	SSA/ACO	Y	Y	N	Y	N	–
(Germi et al., 2018)	Y	N	3	GA/APF	Y	Y	N	Y	N	–
(Gharajeh & Jond, 2022)	Y	N	2	FL/NN	Y	Y	Y	Y	N	GM
(Godio et al., 2021)	N	Y	2	BNN	Y	N	N	Y	N	GM
(Gonzalez et al., 2017)	Y	N	2	CD	Y	N	N	Y	N	GM
(Guo et al., 2019)	N	Y	2	Dijkstra	Y	N	N	Y	N	–
(Guo et al., 2021)	Y	Y	2	FL/NN	Y	Y	N	Y	N	GM
(Guruprasad & Ranjitha, 2021)	Y	N	2	CD	Y	N	N	Y	Y	GM
(Hao et al., 2021)	Y	Y	2	ABC	Y	N	N	Y	N	–
(He et al., 2022)	Y	Y	2	A*	Y	N	N	Y	N	GM
(Huang, 2018a)	Y	Y	2	DWA	Y	N	N	Y	N	GM
(Huang, 2018b)	Y	N	2	Dijkstra	Y	N	N	N	N	–
(Islam et al., 2012)	Y	N	2	RRT	Y	N	N	Y	N	GM
(Jafarzadeh et al., 2018)	Y	N	2	–	–	–	Y	Y	N	GCM
(Jamshidi et al., 2021)	N	Y	3	GWO	Y	N	N	Y	N	–
(Jarray et al., 2022)	Y	N	3	GWO	Y	N	N	Y	N	–
(Ji & Liu, 2022)	Y	N	2	ACO	Y	N	N	Y	N	–
(Jiao et al., 2018)	Y	Y	2	ACO	Y	N	N	Y	N	GM
(Jung et al., 2019)	Y	N	2	CD	Y	N	N	Y	N	GCM
(Karaman & Frazzoli, 2011)	Y	N	2	RRT	Y	N	N	Y	N	GM
(Khan et al., 2021)	N	Y	3	BNN	Y	N	N	Y	N	GM
(Kiani et al., 2021)	Y	N	3	GWO	Y	N	N	Y	N	GM
(Lamini et al., 2018)	Y	N	2	GA	Y	N	N	Y	N	GM
(Li et al., 2020)	Y	N	2	FA	Y	N	N	Y	N	GM
(Li, 2015)	N	Y	2	FL	Y	N	N	Y	Y	–
(Li, 2021)	Y	N	2	SSA	Y	N	N	Y	N	GM
(Li et al., 2022a)	Y	N	2	RRT	Y	N	Y	Y	N	GM
(Li et al., 2021)	Y	N	2	ACO	Y	N	Y	Y	N	GM
(Li et al., 2022b)	Y	Y	2	A*/DWA	Y	Y	N	Y	Y	GM
(Liang et al., 2018)	Y	Y	2/3	–	–	–	N	Y	Y	GCM
(Liu et al., 2021a)	Y	N	3	SSA	Y	N	N	Y	N	–
(Liu et al., 2021b)	Y	N	2	GA/ABC	Y	Y	Y	Y	N	–
(Liu et al., 2022)	Y	N	2	APF/FL	N	Y	N	Y	N	GM
(Liu et al., 2021)	Y	Y	2	GWO	Y	N	N	Y	N	GM
(Liu, 2018)	Y	Y	2	APF	Y	N	N	Y	Y	GM
(Liu et al., 2021c)	N	Y	3	SSA/BNN	Y	Y	Y	Y	N	GM
(Liu & Li, 2022)	Y	Y	2	GWO	Y	N	N	Y	N	GM
(Liu et al., 2021d)	Y	Y	2	APF	Y	N	N	Y	N	–
(Long et al., 2020)	Y	N	2	BFO/A*	Y	Y	N	Y	N	GM
(Long et al., 2021)	Y	N	2	BFO	Y	N	N	Y	N	GM
(Luo et al., 2016)	N	Y	2	BNN	Y	N	N	Y	Y	GM
(Luo et al., 2019b)	Y	N	2	APF	Y	N	Y	Y	N	–

(continued on next page)

Table 25 (*continued*)

Authors	Path planning of		Dimension	Basic Algorithm	Algorithm		Path smooth	Test of		Environmental modeling method
	Global	Local			Optimization	Fusion		Simulation	Actual	
(Mahmud et al., 2019)	Y	N	2	GA	Y	N	N	Y	N	–
(Martins et al., 2022)	Y	N	2	A*	Y	N	Y	Y	N	–
(Miao et al., 2021)	Y	N	2	ACO	Y	N	N	Y	N	GM
(Muni et al., 2021)	Y	N	2	BFO	Y	N	N	Y	Y	–
(Muntasha et al., 2021)	Y	Y	2	ABC	Y	N	N	Y	N	–
(Nazarahari et al., 2019)	Y	N	2	APF/GA	Y	Y	Y	Y	N	–
(Ni & Yang, 2011)	Y	Y	2	BNN	Y	N	N	Y	N	–
(Ni et al., 2017)	N	Y	3	BNN	Y	N	N	Y	N	GM
(Nian, 2021)	N	Y	2	A*	Y	N	Y	N	Y	GM
(Ntakolia & Lyridis, 2021)	N	Y	2	FL/RS	Y	Y	N	Y	N	GM
(Oršulić et al., 2019)	Y	N	2	CD	Y	N	Y	N	Y	GM
(Palmieri et al., 2016)	Y	N	2	RRT	Y	N	Y	Y	Y	GM
(Pang et al., 2021)	Y	N	2	APF	Y	N	N	Y	N	TM
(Panwar & Deep, 2021)	Y	N	2	GWO	Y	N	N	Y	N	–
(Park et al., 2015)	Y	Y	2	CD	Y	N	Y	Y	N	GM
(Patle et al., 2018)	Y	Y	2	FA	Y	N	N	Y	Y	–
(Phung & Ha, 2021)	Y	N	3	PSO	Y	N	N	Y	Y	–
(Qi et al., 2020)	N	Y	2	RRT	Y	N	Y	Y	Y	GM
(Qian et al., 2020)	Y	N	2	RRT	Y	N	N	Y	N	GM
(Qiuyun et al., 2021)	Y	N	2	PSO/GA	Y	Y	N	Y	N	–
(Rahmaniar & (Rakhmania, 2022)	N	Y	2	GA	Y	N	Y	Y	N	–
(Ravankar et al., 2020)	Y	Y	2	PRM/APF	Y	Y	N	Y	Y	GCM
(Ryu & Park, 2019)	Y	N	2	RRT	Y	N	N	Y	N	GM
(Sadhu et al., 2018)	Y	N	3	FA	Y	N	N	Y	Y	GM
(Samaniego et al., 2019)	N	Y	3	CD	Y	N	N	Y	N	GM
(Sang et al., 2021)	N	Y	2	APF	Y	N	N	Y	N	–
(Sangeetha et al., 2021a)	N	Y	3	ACO	Y	N	N	Y	N	GM
(Sangeetha et al., 2021b)	N	Y	2	FL/ACO	Y	Y	Y	Y	N	–
(Sepehri & Moghaddam, 2021)	Y	N	3	APF/RRT	Y	Y	N	Y	Y	–
(Shao et al., 2021)	Y	N	2	ACO	Y	N	Y	Y	N	GM
(Sharma et al., 2021)	Y	N	2	CSA	Y	N	N	Y	N	–
(Shin et al., 2021)	N	Y	3	APF/A*	Y	Y	Y	Y	N	GM
(Shivgan & Dong, 2020)	Y	N	2	GA	Y	N	N	Y	N	–
(Song, 2021)	Y	Y	2	FA	Y	N	N	Y	N	–
(Song et al., 2020)	Y	N	3	CSA	Y	N	N	Y	N	–
(Souza et al., 2019)	Y	N	2	Dijkstra	Y	N	N	Y	N	GM
(Stentz, 1997)	Y	Y	2	A*	Y	N	N	Y	N	GM
(Stenzel et al., 2021)	Y	N	2	RS	Y	N	N	Y	N	GM
(Sun, 2017a)	Y	Y	2	RRT	Y	N	N	Y	N	–
(Sun, 2017b)	N	Y	2	BNN	Y	N	Y	Y	N	GM
(Szczepanski & Tarczewski, 2021)	Y	Y	2	ABC /Dijkstra	Y	Y	N	Y	N	GM
(Szczepanski et al., 2022)	Y	Y	2	APF	Y	N	Y	N	Y	–
(Tang, 2019)	Y	Y	2	PSO /GWO	Y	Y	Y	Y	N	GM
(Tao et al., 2021)	Y	N	2	FL/ACO	Y	Y	N	Y	N	GM
(Tao et al., 2022)	Y	N	2	GA	Y	N	Y	Y	N	GM
(Wang et al., 2021a)	N	Y	2	APF/NN /FL	N	Y	N	Y	Y	GM
(Wang et al., 2022a)	Y	N	2	RRT	Y	N	Y	Y	N	GM
(Wang et al., 2021b)	Y	Y	2	FL/APF	Y	Y	Y	Y	N	–
(Wang et al., 2021c)	Y	N	2	ACO	Y	N	Y	Y	N	GM
(Wang et al., 2021d)	Y	Y	3	BFO	Y	N	N	Y	N	–
(Wang et al., 2022b)	Y	N	2	A*	Y	N	N	Y	N	GM
(Wang et al., 2022c)	Y	N	2	A*	Y	N	Y	Y	N	GM
(Wang et al., 2022d)	Y	N	3	BFO	Y	N	Y	Y	N	–
(Wei et al., 2019)	Y	N	2	Dijkstra	Y	N	N	N	N	–
(Wu, 2021)	Y	N	2	CSA	Y	N	N	Y	N	–
(Wu et al., 2022)	Y	Y	2	A*/DWA	Y	Y	Y	Y	N	GM
(Wu et al., 2021a)	Y	N	2	SSA/GA	Y	Y	N	Y	N	–
(Wu et al., 2021b)	Y	Y	3	PSO	Y	N	N	Y	N	GM
(Xie, 2016)	Y	N	2	FL/APF	Y	Y	N	Y	Y	–
(Xue & Shen, 2020)	Y	N	2	SSA	N	N	N	Y	N	–
(Xu et al., 2021)	Y	N	2	GA	Y	N	N	N	N	–
(Xue et al., 2021)	Y	N	2	ACO	Y	N	Y	Y	N	GM
(Xu et al., 2020)	Y	N	2	RRT	Y	N	Y	Y	N	GM
(Yang et al., 2018)	N	Y	2	BNN	Y	N	N	Y	N	GM
(Yang et al., 2021)	N	Y	3	RS	Y	N	N	Y	Y	TM
(Yang et al., 2022)	Y	Y	2	ACO	Y	N	Y	Y	N	GM
(Yan, 2021)	Y	N	2	ACO	Y	N	N	Y	N	GM
(Yan & Zhao, 2021)	N	Y	3	BNN	Y	N	N	Y	N	GM

(*continued on next page*)

Table 25 (*continued*)

Authors	Path planning of		Dimension	Basic Algorithm	Algorithm		Path smooth	Test of		Environmental modeling method
	Global	Local			Optimization	Fusion		Simulation	Actual	
(Yi et al., 2016)	N	Y	3	BNN	Y	N	N	Y	N	–
(Yu & Ge, 2017)	Y	N	2	Dijkstra	Y	N	N	Y	N	–
(Yu et al., 2018)	Y	Y	2	Dijkstra	N	N	Y	Y	Y	–
(Yu et al., 2021)	Y	N	3	SSA/PSO	Y	Y	Y	Y	N	–
(Zagradjanin et al., 2021)	N	Y	2	FL/A*	Y	Y	N	Y	N	GM
(Zhang et al., 2015)	Y	Y	2	DWA/FL	Y	Y	Y	Y	Y	GCM
(Zhang et al., 2020a)	Y	N	2	PSO	Y	N	Y	Y	Y	GM
(Zhang et al., 2021a)	Y	Y	3	CSA	Y	N	Y	Y	Y	–
(Zhang, 2021b)	N	Y	2	APF	Y	N	N	Y	N	–
(Zhang et al., 2016)	Y	N	2	GWO	Y	N	N	Y	N	–
(Zhang et al., 2020b)	Y	N	2	ACO	Y	N	Y	Y	N	GM
(Zhang et al., 2021c)	Y	N	2	ACO	Y	N	N	Y	N	GM
(Zhang et al., 2021d)	Y	N	2	APF	Y	N	N	Y	N	GM
(Zhang et al., 2022a)	Y	N	3	FA/GA	N	Y	N	Y	N	GM
(Zhang et al., 2021e)	Y	N	3	GWO	Y	N	N	Y	N	–
(Zhang et al., 2018a)	Y	Y	2	A*	Y	N	Y	N	Y	GM
(Zhang, 2020)	N	Y	2	FL/APF	Y	Y	N	Y	Y	GM
(Zhang, 2021)	Y	Y	3	RS	Y	N	N	Y	Y	MR
(Zhang et al., 2018b)	Y	Y	2	Dijkstra	Y	N	N	Y	Y	GM
(Zhang et al., 2022b)	Y	N	2	SSA	Y	N	Y	Y	N	GM
(Zhenyang & Wei, 2022)	Y	Y	2	A*/DWA	Y	Y	Y	Y	Y	GM
(Zheyi & Bing, 2021)	Y	N	2	APF/FL	Y	Y	N	Y	N	–
(Zhou et al., 2018)	Y	Y	3	CD	Y	N	N	Y	N	GM
(Zhu & Yang, 2021)	Y	N	3	BNN	Y	N	N	Y	N	GM
(Zhu et al., 2021)	Y	N	2	ACO /GWO	Y	Y	N	Y	N	GM
(Ziegler et al., 2008)	Y	Y	2	A*	Y	N	Y	Y	Y	TM
Zou et al. (2019)	Y	N	2	PRM	Y	N	N	Y	N	GCM
(Zou, 2005)	Y	Y	3	A*/GA	Y	Y	N	Y	N	MR

Here Y stands for "Yes", N stands for "No" and "-" stands unknown.

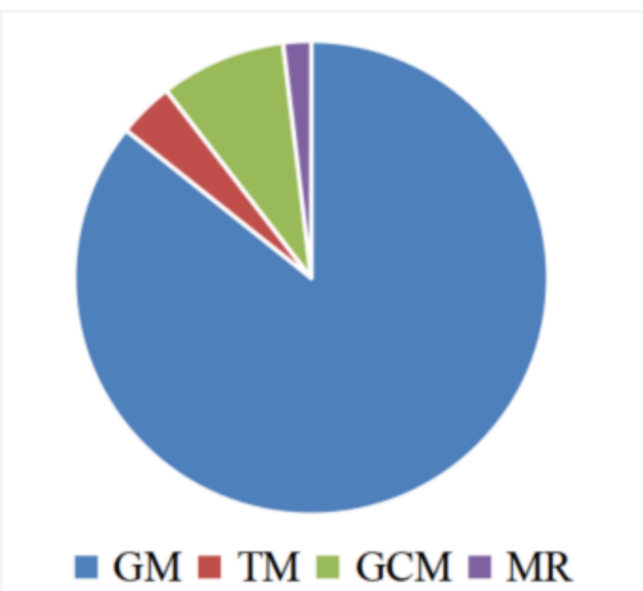


Fig. 14. Data statistics of environment modeling methods for mobile robot.

GWO for solving complex discrete traveler problems. And the 2-optimization algorithm was incorporated in it to improve the performance of the algorithm. Cheng et al. (2021) combined the traditional particle swarm optimization algorithm with GWO, chaos, and adaptive inertia weights to solve the problems of premature convergence and poor global search capability of the particle swarm optimization algorithm, improving the convergence speed of the algorithm and that of path search.

3.2.9. SSA (sparrow search algorithm)

SSA is a swarm intelligence optimization algorithm proposed by Xue & Shen (2020) in 2020 based on sparrows' behavior of foraging and escaping predators (pseudo-code as shown in Table 22).

SSA is a new algorithm with the qualities of better search capability and fast convergence. In the last two years, the optimization for SSA has been mainly focused on:

(1) Smoothed paths. Zhang et al. (2022b) performed path smoothing for turning points on the basis of the traditional algorithm, a new neighborhood search strategy was used to improve the fitness value of the global optimal individual and a new position update function was used to speed up the convergence rate. Liu et al. (2021c) used SSA to find a series of nodes, and then used b-sample curves to fit these nodes, thus achieving path smoothing. Yu et al. (2021) proposed a sparrow particle swarm algorithm for UAV path planning. Three spline interpolation was used to the path nodes, which improved the smoothness of the path and the smoothness of the UAV flight trajectory, making it more applicable to the actual UAV flight trajectory.

(2) Optimization search methods. In order to improve the search capability of SSA in the field of UAV 3D path planning, Liu et al. (2021a) transformed the route planning problem into a multidimensional function optimization problem, introduced chaos strategy to enhance the diversity of the algorithm population, and used adaptive inertia weights to balance the convergence speed and search capability of the algorithm. Finally, a Cauchy-Gaussian mutation strategy was used to improve the ability of the algorithm to solve the stagnation problem. Wu et al. (2021a) proposed a greedy genetic SSA based on a sine and cosine search strategy and applied to the traveler problem. Li (2021) introduced a K-medoids clustering method to dynamically classify populations based on SSA, speeding up the information exchange between populations. In addition, an adaptive weighting factor was introduced to make the algorithm search more detailed and extensive, which improved its optimization capabilities. Gao et al. (2021) proposed the ant-sparrow algorithm based on the same number of constraint sets of ants and sparrows, which combined the qualities of a better initial solution of the

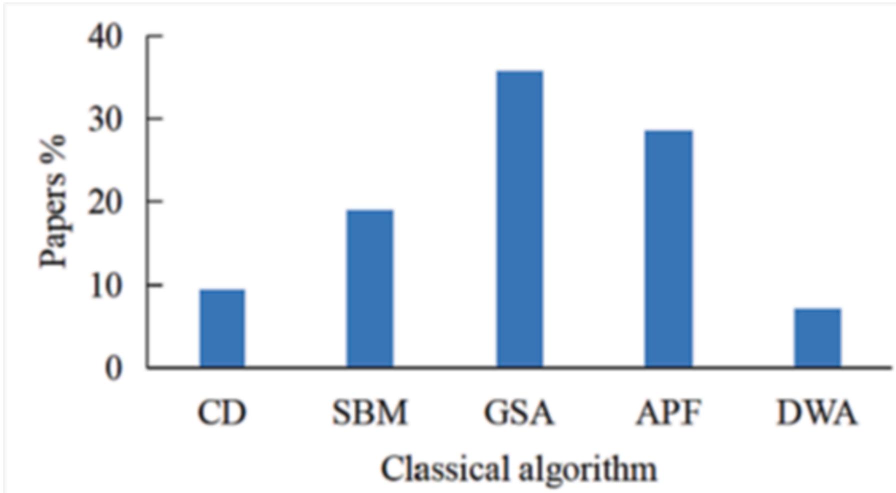


Fig. 15. Data statistics of classical path planning algorithms based on literatures in this paper.

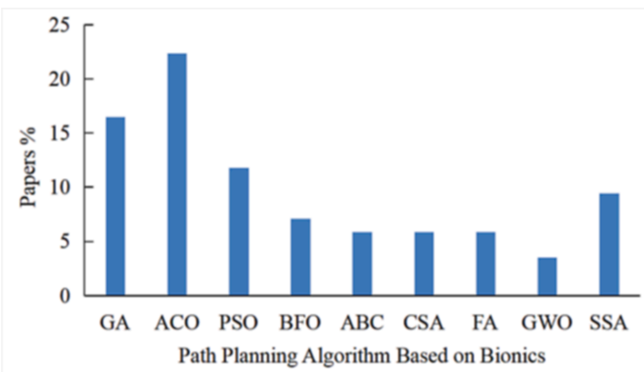


Fig. 16. Data statistics of bionic path planning algorithms based on literatures in this paper.

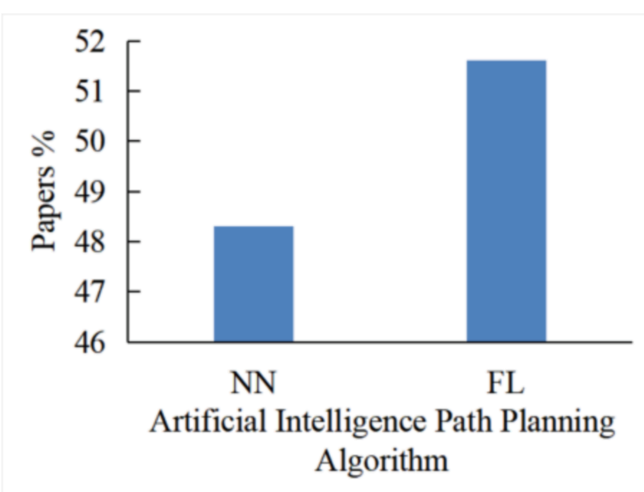


Fig. 17. Data statistics of artificial intelligence path planning based on literatures in this paper.

ant colony optimization algorithm and the fast convergence of SSA. After limited initial iterations, the ant colony algorithm was transferred into SSA. Yu et al. (2021) proposed that SPSA algorithm can change the position update of the finders, enhance the influence of the starting line on the path search, and significantly reduce the blind search.

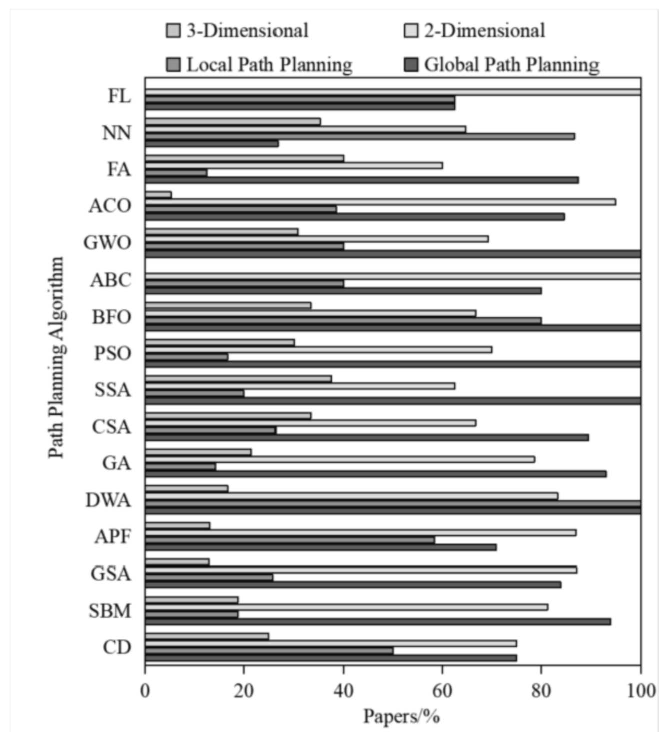


Fig. 18. Data statistics of the application environment of path planning algorithms based on literatures in this paper.

Table 23 clearly shows the difference between the above-mentioned bionics-based path planning algorithms (Tan et al., 2021).

3.3. AI (artificial intelligence) algorithm

AI is a study of making computers to simulate thought process and intelligent behaviors of human beings (e.g., learning, reasoning, thinking, planning, etc.). The three cornerstones of artificial intelligence are algorithms, data, and computing capability. In this paper, we focus on two popular AI algorithms for path planning of mobile robots, including bioinspired neural network algorithm and fuzzy control algorithm.

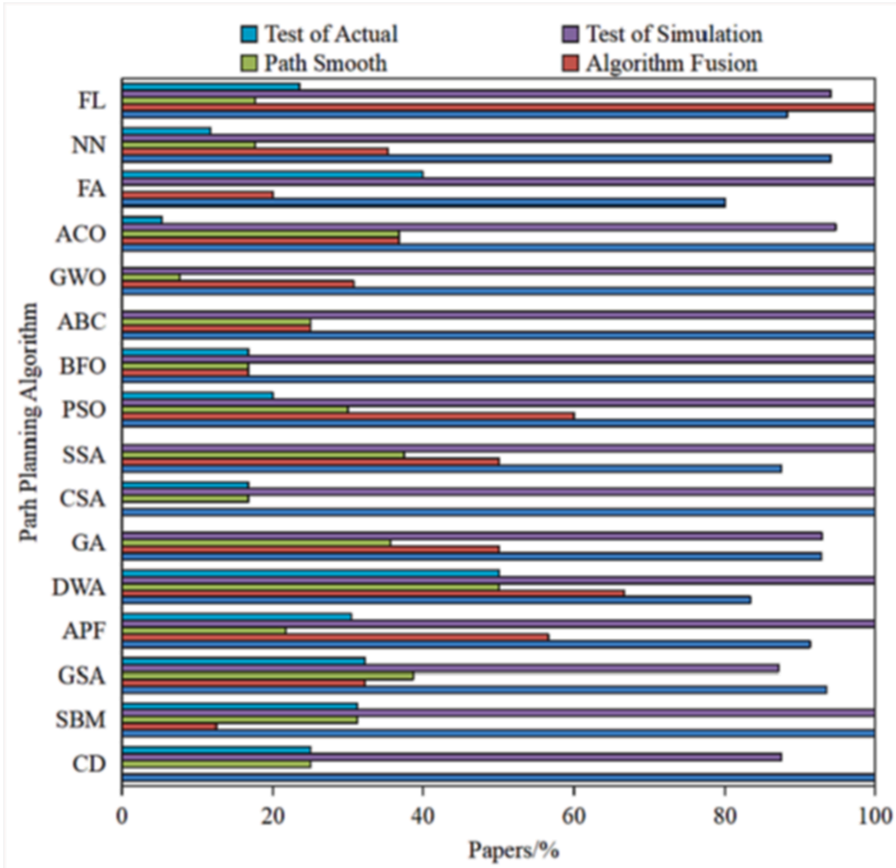


Fig. 19. Data statistics of optimization methods and experimental methods of path planning algorithms based on literatures in this paper.

3.3.1. BNN (bioinspired neural network)

Meng & Yang (Luo et al., 2019a) firstly introduced the BNN method to solve the path planning problem of mobile robots in 1998. Compared with other NN algorithms, BNN algorithms do not need to be learned (Ni & Yang, 2011). The dynamic activation value function of each neuron cell can be solved through information transfer between neurons. And by solving the dynamic activation potential function of neurons in real time, the feasible path from the initial position to the target position along the direction of increasing potential value can be obtained as the planning path (Luo, 2021). At present, the structure of BNN contains input layer, hidden layer, and output layer (Zhao, 2020)..

A BNN-based path planning method for multi-robot capture was proposed by Chen & Zou (2021). They constructed a dynamic alliance strategy to achieve the interaction of multiple robots, constructed a target tracking strategy based on bio-inspired neural networks to dynamically guide all robots in the coalition for tracking, and used the formation strategy to capture the target. Yi et al. (2016) integrated the advantages and features of BNN with the self-organizing feature map to handle the task allocation of large batches of robots in 3D dynamic environments. Based on the improved BNN algorithm, a real-time collision-free path planning method for descaling robots in a naval environment was proposed by Sun (2017b). Ni et al. (2017) proposed an improved dynamic BNN algorithm to handle real-time path planning for autonomous underwater vehicle in various 3D underwater environments. Yang et al. (2018) proposed an improved BNN-based "active obstacle avoidance strategy" pedestrian position prediction method to plan collision-free complete coverage path planning trajectories for mobile robots. Luo et al. (2016) proposed a BNN-based dynamics approach to handle the complete coverage path planning of multiple robots. Wang et al. (2019) used BNN techniques to assist the robot in patrolling the unknown working environments, and used the fast R-CNN (regions with convolutional neural networks) approach to discover scattered nails and screws in real time, so that the robot can automatically recover nails and screws. A shortest path deduction algorithm based on neural network was designed for underwater robots by Zhu & Yang (2021). Yan & Zhao (2021) constructed the GBNN (glasius bio-inspired neural network) model with a topological structure, which represented the search environment. Gaussian mixture model was used to extract high-value subregions and quantify future detection rewards. The external excitatory input to the neurons of the GBNN can be directly introduced as a way to improve the search capability. Godio et al. (2021) described an approach based on bio-inspired neural network, which can solve the coverage planning problem for unmanned aerial vehicle formations exploring critical areas. Khan et al. (2021) used the ZNNBAS (zeroing neural network with beetle antennae search) to solve for mobile home robots, further improving the computing ability and efficiency of the system. ZNNBAS is a nature-inspired meta heuristic algorithm inspired by the food-seeking nature of beetles.

3.3.2. FL (fuzzy logic)

Zadeth (1965) proposed the concept of FL in 1965. Based on real-time information from sensors, FL regards current environmental obstacle information as input of the fuzzy controller, analyzes the desired speed by simulating human driving experience, constructs a planning information table, and finally controls the mobile robot to complete local path planning in an unstructured environment. With better real-time performance, FL is less influenced by the external environment and is suitable for path planning in unknown environments (Li, 2015).

However, FL also has some inherent defects. For example, fuzzy inference rules should be formulated in advance based on experts' experience, which is closely related to the advantages and disadvantages of path; the optimal path depends on fuzzy inference rules; as the obstacle information increases, the inference rules or fuzzy tables will be sharply dilated, which affects the real-time performance of the algorithm; when the environment changes, the existing rules may not be applicable to the new environment, resulting in the failure to choose the correct path; the obstacle avoidance strategy is not intelligent and cannot flexibly cope with dynamic obstacles. Many scholars proposed to improve the online learning capability and accuracy of FL by combining the FL algorithm with others (Xie, 2016). Zagradjanin et al. (2021), used D*lite algorithm in global path planning and used FL in local path planning, and introduced this method into collaborative multi-robot operations in complex environments. Ntakolia & Lyridis (2021) proposed swarm intelligence graph-based pathfinding algorithm (SIGPA), which aims to improve the performance of the SIGPA algorithm, cope with multiple objectives and produce high-quality solutions by integrating fuzzy logic. Gharajeh & Jond (2022) proposed a collision-free autonomous navigation path planning method for wheeled mobile robots based on adaptive neuro-fuzzy inference system (ANFIS). And experimental results showed that the paths found by this method were about 30% shorter than those found by other methods. Guo et al. (2021) proposed a new algorithm based on a combination of long and short-term memory neural networks, fuzzy logic control and reinforcement learning control. And low-dimensional input fuzzy logic control (FL) algorithm was applied to collect training data and the transfer learning method was applied to learn the basic capabilities. Wang et al. (2021b), proposed hierarchical path planning (HPP) scheme combining global and local tasks for unmanned surface vehicles by considering fuzzy artificial potential field (FAPF). Wang et al. (2021a) introduced NN algorithm into FAPF. The training samples of the neural network consists of the coordinates of the target and the obstacle, as well as the motion direction of the robot corresponding to this position relationship, which can further improve the path optimization capability of the mobile robots. AlKhlidi et al. (2021) proposed FPSO algorithm combining PSO with FL that can improve the problem of PSO algorithm falling into local optimal solutions. Sangeetha et al. (2021b) proposed fuzzy gain-based dynamic ant colony optimization (FGDACO) for dynamic path planning, which effectively plans collision-free and smooth paths with the shortest length and time. Tao et al. (2021) introduced a pheromone gain-based function to the path planning process in order to effectively utilize the pheromone update mechanism of the ant colony system. A new pheromone update rule was proposed, which used fuzzy control to change the magnitude of the pheromone heuristic factor and the desired heuristic factor, as well as adjust the evaporation rate in stages. This method converged quickly, ensuring the global search capability.

Table 24 clearly shows the difference between BNN and FL.

4. Conclusion and prospection

4.1. Conclusion

Based on the statistics of the research-oriented literature covered in this paper, we made Table 25. In order to explore the development trend of path planning algorithms, we performed a data-based analysis of Table 3 and obtained Figs. 14–19.

Among the 105 papers regarding environment modeling, the percentages of models using grid method, topology method, geometrical measurers, and hybrid methods are 85.71%, 3.81%, 8.57%, and 1.90%, respectively. In Fig. 14, it can be intuitively found that the grid method, as the mainstream modeling method in path planning of mobile robots, has been irreplaceable in the past and even in the future.

In recent years, many classical path planning algorithms have been proposed. CD, SBM and GSA have contributed more to global path planning than local path planning, while APF's contribution to global path planning is almost equal to that to local path planning. DWA is a path planning algorithm for dynamic environment. CD is seldom fused with other algorithms, and researchers usually adopt the approach of optimizing the CD algorithm itself to deal with its shortcomings. SBM, GSA and APF are mainly used to improve the algorithm itself, but compared with GSA, APF and DWA is the easiest to fuse with other algorithms. Among the bionics-based path planning algorithms, there are fewer studies for local path planning than global path planning. However, compared with classical path planning, these algorithms response quickly and are more suitable to be optimized for local path planning. We also find that there are a large proportion of research papers on ACO, GWO, GA and PSO which have shorter search time and stronger search capability. For these algorithms, researchers are keen on optimizing them for a particular defect rather than combining them with other algorithms. But PSO, SSA, and GA are more easily fused with other algorithms. In the field of AI algorithms, the proportion of researches on NN and FL is basically equal. What's more, AI algorithm is more often used in local path planning due to its higher environment exploration capability and more flexible operation capability. FL is rarely used for path planning in three-dimension environments because it is more complex to design and requires larger memory. But FL is often fused with other algorithms, or it is used to optimize the certain defect of other algorithms. In Fig. 19, it can be seen that bionics-based and artificial intelligence algorithms contribute more to local path planning than classical algorithms. Some intrinsic defects of classical algorithm, such as local optimal solutions, lack of intelligence, and low computational intensity, cause the failure of performing real-time path planning.

This paper reviews the path planning techniques of mobile robots based on a large amount of literature and categories the research-oriented literature. The conclusions are as follows.

(1) The environment modeling for mobile robots usually uses gird method that is easy to implement and extend to three-dimension environment.

(2) Classical path planning algorithms are often used to perform global path planning. They need small memory occupation without expensive environmental monitoring systems, becoming very popular in low-cost mobile robots.

(3) Artificial intelligence algorithms are most preferred in local path planning because they are trained to learn with higher intelligence and better to handle uncertainty in the environment.

(4) There is less research on path planning in three-dimension environments than two-dimension environments.

(5) Researchers are keen on improving a particular algorithm rather than fusing different algorithms.

(6) The experiment verification method used in most of the papers is simulating on a computer rather than applying it in real environments of mobile robots.

(7) One third of the researchers considered path smoothing techniques.

4.2. Prospection

With the continuous development of computer science, mobile robots are widely used in industry, agriculture, service industry and other fields, which requires that mobile robot systems have good real-time and robustness. Therefore, in the future path planning technology, in addition to research and discovery of new path planning algorithms, the following aspects deserve attention:

Environment modeling technology is combined with path planning algorithm. When a mobile robot is in a complex 2D or even 3D continuous dynamic environment, the function of the algorithm is limited. The combination of good modeling technology and path planning algorithm will become a method to solve this problem.

The problem of combining global path planning with local path planning. Global path planning refers to the condition that all the data in the environmental map are known, and the safety node and safety area, as well as the optimal driving path, are judged by combining the real-time information obtained with the feedback control. The global path can only be roughly planned, but mobile robot driving is a very complex problem. Especially after the mobile robot is applied to military industry, agriculture, service industry and other occasions with high real-time requirements, not only the initial environmental data processing, but also the dynamic factors such as the state of the robot when moving and the characteristics of obstacles should be considered. Of course, the width and curvature of the path, Static factors such as the particularity of intersections should also be included. Therefore, the mobile robot constantly interacts with the environment to obtain local paths, and enough local paths constitute the global path.

Improvement of advanced path planning algorithm. When the environment changes, traditional algorithms will expose their shortcomings. For example, mobile robots are widely used to deliver meals in many countries, and the scene is densely populated. Researchers can consider adding learning function to the existing algorithm. After continuous learning, the probability of collision of the delivery robot will become smaller and smaller.

Effective combination of path planning algorithms. Each algorithm has inherent defects due to its own limitations in practical application. Through the fusion of two or more algorithms, we can learn from each other's strengths to offset each other's weaknesses. To some extent, it can avoid the defects of a single algorithm.

Design of path planning algorithm for multi machine collaborative work. With the continuous development of science and technology, the problem of multi machine collaborative parallel operation and path conflict has been gradually concerned. How to realize the path planning of multi robot collision free operation will become one of the research focuses in the future.

CRediT authorship contribution statement

Lixing Liu: Conceptualization, Methodology, Validation, Writing – original draft. **Xu Wang:** Software, Conceptualization, Supervision. **Xin Yang:** Writing – review & editing, Formal analysis, Supervision, Visualization. **Hongjie Liu:** Investigation, Resources, Data curation. **Jianping Li:** Formal analysis, Funding acquisition, Project administration. **Pengfei Wang:** Software, Data curation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the earmarked fund for CARS (CARS-27) and supported by the Hebei Province Graduate Innovation Funding Project (CXZZBS2023080).

References

Abdallaoui, S., Aglzim, E. H., Chaibet, A., et al. (2022). Thorough review analysis of safe control of autonomous vehicles: Path planning and navigation techniques. *Energies, 15*(4), 1358.

Abdi, M. I. I., Khan, M. U., Güneş, A., et al. (2020). Escaping Local Minima in Path Planning Using a Robust Bacterial Foraging Algorithm. *Applied Sciences, 10*(21), 7905.

Albina, K., & Lee, S. G. (2019). Hybrid stochastic exploration using grey wolf optimizer and coordinated multi-robot exploration algorithms. *IEEE Access, 7*, 14246–14255.

Alkhlidi, B., Abdulsadda, A. T., & Al, B. A. (2021). Optimal Robotic Path Planning Using Intlligents Search Algorithms. *Journal of Robotics and Control (JRC), 2*(6), 519–526.

Alyasin, A., Abbas, E. I., & Hasan, S. D. (2019). An efficient optimal path finding for mobile robot based on dijkstra method. In *2019 4th Scientific International Conference Najaf (SICN)* (pp. 11–14). IEEE.

An, Y. Y., & Zeng, X. D. (2001). Optical sensing and measurement. *Beijing Electronic Industry Press, 293*, 294.

Balado, J., Díaz-Vilariño, L., Arias, P., et al. (2019). Point clouds for direct pedestrian pathfinding in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing, 148*, 184–196.

Bao J.S., Zhang M.Y., Ge S.R., et al., Underground Driverless Path Planning of Trackless Rubber Tyred Vehicle Based on Improved A * and Artificial Potential Field Algorithm[J/OL]. Journal of China Coal Society: 1-14[2022-03-22]. DOI:10.13225/j.cnki.jccs.XR21.1717.

Brunner M., Brüggemann B., Schulz D., Hierarchical rough terrain motion planning using an optimal sampling-based method. 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013: 5539-5544.

Bullnheimer B., Hartl R.F., Strauss C. A new rank based version of the Ant System. A computational study[J]. 1997.

Chai, R., Tsourdos, A., Savvaris, A., et al. (2021). Solving constrained trajectory planning problems using biased particle swarm optimization. *IEEE Transactions on Aerospace and Electronic Systems, 57*(3), 1685–1701.

Chang, Z., Wu, H., Sun, Y., et al. (2022). RGB-D Visual SLAM Based on Yolov4-Tiny in Indoor Dynamic Environment. *Micromachines, 13*(2), 230.

Challita, U., Saad, W., & Bettstetter, C. (2018). Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs. In *2018 IEEE international conference on communications (ICC)* (pp. 1–7). IEEE.

Chen, G., Luo, N., Liu, D., et al. (2021a). Path planning for manipulators based on an improved probabilistic roadmap method. *Robotics and Computer-Integrated Manufacturing, 72*, Article 102196.

Chen, Y. G. (2020). Local Tracking and Global Localization of Mobile Robot based on Sensor Fusion of Vision and LiDAR[D]. *Guangdong University of Technology*. https://doi.org/10.27029/d.cnki.ggdgu.2020.000295

Chen, Y., Bai, G., Zhan, Y., et al. (2021b). Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning. *IEEE Access, 9*, 40728–40742.

Chen, Y., & Zhou, X. (2021). Path planning of robot based on improved ant colony algorithm in computer technology. *Journal of Physics: Conference Series IOP Publishing, 1744*(4), Article 042092.

Chen, Z., Zhou, J., Sun, R., et al. (2021c). A new evolving mechanism of genetic algorithm for multi-constraint intelligent camera path planning. *Soft Computing, 25*(7), 5073–5092.

Chen, Z., & Zou, A. C. (2021). Research on multi-robot cooperative roundup based on biological heuristic neural network. *Electronic Measurement Technology, 44*(10), 82–90. https://doi.org/10.19651/j.cnki.emt.2106018

Chengli, F. A. N., Qiang, F. U., Guang, Z. L., et al. (2018). Hybrid artificial bee colony algorithm with variable neighborhood search and memory mechanism. *Journal of Systems Engineering and Electronics, 29*(2), 405–414.

Cheng, X., Li, J., Zheng, C., et al. (2021). An improved PSO-GWO algorithm with chaos and adaptive inertial weight for robot path planning. *Frontiers in Neurorobotics*, 15.

Chien, R. T., Zhang, L., & Zhang, B. (1984). Planning collision-free paths for robotic arm among obstacles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 1*, 91–96.

Chou, J. S., & Ngo, N. T. (2017). Modified firefly algorithm for multidimensional optimization in structural design problems. *Structural and Multidisciplinary Optimization, 55*(6), 2013–2028.

Chu, K. X., Chang, T. Q., Wang, Q. D., et al. (2021). Path Planning Method of Ant Colony Algorithm Based on Effective Turning Point and Shortest-minimum Path. *Transactions of the Chinese Society for Agricultural Machinery, 52*(12), 400–407.

Chu, X., Hu, G., Niu, B., et al. (2016). An superior tracking artificial bee colony for global optimization problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2712–2717). IEEE.

Dai, X., Long, S., Zhang, Z., et al. (2019). Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Frontiers in Neurorobotics, 13*, 15.

Daniel, K., Nash, A., Koenig, S., et al. (2010). Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research, 39*, 533–579.

Daraei, M. H. (2018). *Tightly-coupled lidar and camera for autonomous vehicles[M]*. Santa Cruz: University of California.

Debnath, S. K., Omar, R., Bagchi, S., et al. (2021). Different cell decomposition path planning methods for unmanned air vehicles-A review. In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019* (pp. 99–111). Singapore: Springer.

Detone D., Malisiewicz T., Rabinovich A., Toward Geometric Deep SLAM[J]. 2017.

Dhal, K. G., Iqbal, Q. M., & Das, S. (2016). Development of firefly algorithm via chaotic sequence and population diversity to enhance the image contrast. *Natural Computing, 15*(2), 307–318.

Ding, X. Q., Du, Z. Y., Lu, Y. Q., et al. (2016). Visual trajectory planning for mobile robots based on hybrid artificial potentialfield. *Journal of Zhejiang University(Engineering Science), 50*(07), 1298–1306.

Dorigo M., Maniezzo V., Colorni A., The ant system: An autocatalytic optimizing process [J]. 1991.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *Bio Systems, 43*(2), 73–81.

Du, W. R., Wang, X. Y., Tian, T., et al. (2021). Artificial potential field path planning algorithm for unknown environment and dynamic obstacles. *Computer Science, 48*(02), 250–256.

Esposito, J. M., & Wright, J. N. (2019). Matrix completion as a post-processing technique for probabilistic roadmaps. *The International Journal of Robotics Research, 38*(2/3), 388–400.
Fan, Y., Sun, X., Wang, G., et al. (2021). Collision avoidance controller for unmanned surface vehicle based on improved cuckoo search algorithm. *Applied Sciences, 11*(20), 9741.
Fei W., Ziwei W., Meijin L., Robot Path Planning Based on Improved Particle Swarm Optimization. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021: 887-891.
Feng, S., Qian, Y., & Wang, Y. (2021). Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 235*(14), 3416–3430.
Ferguson, D., & Stentz, A. (2006). Using interpolation to improve path planning: The Field D* algorithm. *Journal of Field Robotics, 23*(2), 79–101.
Fink, W., Baker, V. R., Brooks, A. J. W., et al. (2019). Globally optimal rover traverse planning in 3D using Dijkstra's algorithm for multi-objective deployment scenarios. *Planetary and Space Science, 179*, Article 104707.
Fox, D., Burgard, W., & Thrun, S. (2002). The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine, 4*(1), 23–33.
Gao, Q., Zheng, J., & Zhang, W. (2021). Research on optimization of manned robot swarm scheduling based on ant-sparrow algorithm. *Journal of Physics: Conference Series. IOP Publishing, 2078*(1), Article 012002.
Germi S.B., Khosravi M.A., FesharakiFard R., Adaptive GA-based potential field algorithm for collision-free path planning of mobile robots in dynamic environments. 2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM). IEEE, 2018: 28-33.
Gharajeh, M. S., & Jond, H. B. (2022). An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system. *Ain Shams Engineering Journal, 13*(1), Article 101491.
Godio, S., Primatesta, S., Guglieri, G., et al. (2021). A Bioinspired neural network-based approach for cooperative coverage planning of UAVs. *Information, 12*(2), 51.
Gomez C., Fehr M., Millane A., et al., Hybrid topological and 3d dense mapping through autonomous exploration for large indoor environments. 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 9673-9679.
Gonzalez R., Kloetzer M., Mahulea C., Comparative study of trajectories resulted from cell decomposition path planning approaches. 2017 21st International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2017: 49-54.
Graeter, J., Wilczynski, A., & Lauer, M. (2019). LIMO: Lidar-monocular visual odometry. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
Gregorio, D. D., & Stefano, L. D. (2017). SkiMap: An Efficient Mapping Framework for Robot Navigation. *IEEE*. IEEE.
Guivant, J., Nebot, E., & Durrant-Whyte, H. (2000). Simultaneous localization and map building using natural features in outdoor environments. *Intelligent Autonomous Systems., 6*(1), 581–586.
Guo, D., Wang, J., Zhao, J. B., et al. (2019). A vehicle path planning method based on a dynamic traffic network that considers fuel consumption and emissions. *Science of the Total Environment, 663*, 935–943.
Guo, N., Li, C., Gao, T., et al. (2021). A fusion method of local path planning for mobile robots based on LSTM neural network and reinforcement learning. *Mathematical Problems in Engineering, 2021*.
Guruprasad, K. R., & Ranjitha, T. D. (2021). CPC algorithm: Exact area coverage by a mobile robot using approximate cellular decomposition. *Robotica, 39*(7), 1141–1162.
Hao, W., Luo, B., & Zhang, Z. (2021). Application study of UAV path planning based on the balanced search factor artificial bee colony algorithm. *Journal of Physics: Conference Series. IOP Publishing, 2083*(3), Article 032064.
He, Z., Liu, C., Chu, X., et al. (2022). Dynamic anti-collision A-star algorithm for multi-ship encounter situations. *Applied Ocean Research, 118*, Article 102995.
Howden, W. E. (1968). The sofa problem. *The Computer Journal, 11*(3), 299–301.
Huang, C. (2018a). *Study on Path Planning and Location of Mobile Robot Based on Intelligent Optimization Algorithm [2018]*. Dalian Jiaotong University.
Huang Y. Research on the improvement of Dijkstra algorithm in the shortest path calculation. Proceedings of the 2017 4th International Conference on Machinery, Materials and Computer (MACMC 2017), Atlantis Press. 2018b: 745-749.
Islam, F., Nasir, J., Malik, U., et al. (2012). Rrt*-smart: Rapid convergence implementation of rrt* towards optimal solution. In *2012 IEEE international conference on mechatronics and automation* (pp. 1651–1656). IEEE.
Jafarzadeh, H., & Fleming, C. H. (2018). An exact geometry-based algorithm for path planning. *International Journal of Applied Mathematics and Computer Science, 28*(3).
Jain, M., Saihjpal, V., Singh, N., et al. (2022). An Overview of Variants and Advancements of PSO Algorithm. *Applied Sciences, 12*(17), 8392.
Jamshidi, V., Nekoukar, V., & Refan, M. H. (2021). Real time UAV path planning by parallel grey wolf optimization with align coefficient on CAN bus. *Cluster Computing, 24*(3), 2495–2509.
Jarray, R., Al-Dhaifallah, M., Rezk, H., et al. (2022). Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles. *Sensors, 22*(5), 1826.
Ji, Y., & Liu, B. (2022). Research and Implementation of Robot Path Planning Based on Ant Colony Algorithm. *Journal of Physics: Conference Series. IOP Publishing, 2171*(1), Article 012074.
Jiao, Z., Ma, K., Rong, Y., et al. (2018). A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs. *Journal of Computational Science, 25*, 50–57.
Jung, J. W., So, B. C., Kang, J. G., et al. (2019). Expanded douglas–peucker polygonal approximation and opposite angle-based exact cell decomposition for path planning with curvilinear obstacles. *Applied Sciences, 9*(4), 638.
Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research, 30*(7), 846–894.
Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.
Khan, A. T., Li, S., & Cao, X. (2021). Control framework for cooperative robots in smart home using bio-inspired neural network. *Measurement, 167*, Article 108253.
Kiani, F., Seyyedabbasi, A., Aliyev, R., et al. (2021). 3D path planning method for multi-UAVs inspired by grey wolf algorithms. *Journal of Internet Technology, 22*(4), 743–755.
Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science, 127*, 180–189.
Latombe, J. C. (1991). Approximate cell decomposition. In *Robot Motion Planning* (pp. 248–294). Boston, MA: Springer.
Li, F., Fan, X., & Hou, Z. (2020). A Firefly Algorithm With Self-Adaptive Population Size for Global Path Planning of Mobile Robot. *IEEE Access, 8*, 168951–168964.
Li, H. L. (2020). Improved artificial bee colony algorithm and related application research[D]. *Nanjing University of Posts and Telecommunications*. https://doi.org/10.27251/d.cnki.gnjdc.2020.001231
Li, M. (2014). *Visual-inertial odometry on resource-constrained systems[M]*. Riverside: University of California.
Li, M. (2015). Mobile robot path planning based on fuzzycontrol. *Hebei University of Technology*.
Li, J. (2021). Robot path planning based on improved sparrow algorithm. *Journal of Physics: Conference Series. IOP Publishing, 1861*(1), Article 012017.
Li, Q. Q., Bu, S. Q., & Yang, J. F. (2021). Research review on biological swarm intelligence algorithm in mobile robot path planning. *World Sci-Tech R&D, 43*(05), 535–546. https://doi.org/10.16507/j.issn.1006-6055.2021.04.004
Li, Q., Wang, J., Li, H., et al. (2022a). Fast-RRT*: An improved motion planner for mobile robot in two-dimensional space. *IEEJ Transactions on Electrical and Electronic Engineering, 17*(2), 200–208.
Li, Y., Ming, Y., Zhang, Z., et al. (2021). An adaptive ant colony algorithm for autonomous vehicles global path planning. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 1117–1122).
Li, Y., Li, J., Zhou, W., et al. (2022b). Robot path planning navigation for dense planting red jujube orchards based on the joint improved A* and DWA algorithms under laser SLAM. *Agriculture, 12*(9), 1445.
Liang, X. (2016). Indoor slam for robots based on laser and mono-vision fusion. *Harbin Institute of Technology*.
Liang, X., Meng, G., Xu, Y., et al. (2018). A geometrical path planning method for unmanned aerial vehicle in 2D/3D complex environment. *Intelligent Service Robotics, 11*(3), 301–312.
Lin, H. X., Xiang, D., & Ou, Y. J. (2021). Review of Path Planning Algorithms for Mobile Robots. *Computer Engineering and Applications, 57*(18), 38–48.
Liu, G., Shu, C., Liang, Z., et al. (2021a). A modified sparrow search algorithm with application in 3d route planning for UAV. *Sensors, 21*(4), 1224.
Liu, H., Ge, J., Wang, Y., et al. (2021b). Multi-UAV optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method. *Actuators MDPI, 11*(1), 4.
Liu, H. P., Wang, D. S., Wang, Y. X., et al. (2022). Research of path planning for mobile robots based on fuzzy artificial potential field method. *Control Engineering of China, 29*(01), 33–38. https://doi.org/10.14107/j.cnki.kzgc.20200187
Liu, J. (2019). Research and application of clustering recommendation algorithm based on cuckoo search. *Software Guide, 18*(04), 91–94.
Liu, J., Wei, X., & Huang, H. (2021). An improved grey wolf optimization algorithm and its application in path planning. *IEEE Access, 9*, 121944–121956.
Liu, K. (2018). *Object tracking and path planning of indoor mobile robot based on visual information*. South China University of Technology.
Liu, Q., Zhang, Y., Li, M., et al. (2021c). Multi-UAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network. *IEEE Access, 9*, 124670–124681.
Liu, Y., & Li, X. (2022). A Hybrid mobile robot path planning scheme based on modified gray wolf optimization and situation assessment. *Journal of Robotics, 2022*.
Liu, Z., Yuan, X., Huang, G., et al. (2021d). Two potential fields fused adaptive path planning system for autonomous vehicle under different velocities. *ISA Transactions, 112*, 176–185.
Long, Y., Zuo, Z., Su, Y., et al. (2020). An A*-based bacterial foraging optimisation algorithm for global path planning of unmanned surface vehicles. *The Journal of Navigation, 73*(6), 1247–1262.
Long, Y., Su, Y., Shi, B., et al. (2021). A multi-subpopulation bacterial foraging optimisation algorithm with deletion and immigration strategies for unmanned surface vehicle path planning. *Intelligent Service Robotics, 14*(2), 303–312.
Luo, C., Yang, S. X., Li, X., et al. (2016). Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots. *IEEE Transactions on Industrial Electronics, 64*(1), 750–760.
Lu, C. (2017). *Study on theory and method of multi-objective scheduling problems with controllable processing times*. Huazhong University of Science and Technology.
Luo, M. (2021). Research on path planning and trajectory tracking of mobile robot in complex environment. *University of Electronic Science and Technology of China*. https://doi.org/10.27005/d.cnki.gdzku.2021.000212
Luo M, Hou X, Yang J. Multi-robot one-target 3d path planning based on improved bioinspired neural network[C]//2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing. IEEE, 2019a: 410-413.

Luo, Q., Wang, H. B., Cui, X. J., et al. (2019b). Autonomous mobile robot path planning based on improved artificial potential method. *Control Engineering of China, 26*(06), 1091–1098. https://doi.org/10.14107/j.cnki.kzgc.170907

Mahmud, M. S. A., Abidin, M. S. Z., Mohamed, Z., et al. (2019). Multi-objective path planner for an agricultural mobile robot in a virtual greenhouse environment. *Computers and Electronics in Agriculture, 157*, 488–499.

Ma, F., Shi, J., Yang, Y., et al. (2019). ACK-MSCKF: Tightly-coupled Ackermann multi-state constraint Kalman filter for autonomous vehicle localization. *Sensors, 19*(21), 4816.

Martins, O. O., Adekunle, A. A., Olaniyan, O. M., et al. (2022). An Improved multi-objective a-star algorithm for path planning in a large workspace: Design, Implementation, and Evaluation. *Scientific African, 15*, e01068.

Miao, C., Chen, G., Yan, C., et al. (2021). Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Computers & Industrial Engineering, 156*, Article 107230.

Mirjalili S., Mirjalili S.M., Lewis A., Grey wolf optimizer. Advances in Engineering Software 69, 46-61[J]. 2014.

Mohanty, P. K., Singh, A. K., Kumar, A., et al. (2021). Path planning techniques for mobile robots: A review. In *International Conference on Soft Computing and Pattern Recognition* (pp. 657–667). Cham: Springer.

Mourikis, A. I., & Roumeliotis, S. I. (2007). A multi-state constraint kalman filter for vision-aided inertial navigation. *ICRA, 2*, 6.

Muni, M. K., Parhi, D. R., & Kumar, P. B. (2021). Improved motion planning of humanoid robots using bacterial foraging optimization. *Robotica, 39*(1), 123–136.

Muntasha G., Karna N., Shin S.Y., Performance Analysis on Artificial Bee Colony Algorithm for Path Planning and Collision Avoidance in Swarm Unmanned Aerial Vehicle[C]//2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS). IEEE, 2021: 1-6.

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications, 115*, 106–120.

Ni, J., & Yang, S. X. (2011). Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments. *IEEE Transactions on Neural Networks, 22*(12), 2062–2077.

Ni, J., Wu, L., Shi, P., et al. (2017). A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles. *Computational intelligence and neuroscience, 2017*.

Nian, J. (2021). Indoor map construction and navigation of mobile robot based on multi-sensor fusion. *Beijing University of Posts and Telecommunications*. https://doi.org/10.26969/d.cnki.gbydu.2021.002501

Ntakolia, C., & Lyridis, D. V. (2021). A swarm intelligence graph-based pathfinding algorithm based on fuzzy logic (SIGPAF): A case study on unmanned surface vehicle multi-objective path planning. *Journal of Marine Science and Engineering, 9*(11), 1243.

Oršulić, J., Miklić, D., & Kovačić, Z. (2019). Efficient dense frontier detection for 2-d graph slam based on occupancy grid submaps. *IEEE Robotics and Automation Letters, 4*(4), 3569–3576.

Palmieri L., Koenig S., Arras K.O., RRT-based nonholonomic motion planning using any-angle path biasing[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 2775-2781.

Pang, B., Dai, W., Hu, X., et al. (2021). Multiple air route crossing waypoints optimization via artificial potential field method. *Chinese Journal of Aeronautics, 34*(4), 279–292.

Panwar, K., & Deep, K. (2021). Discrete Grey Wolf Optimizer for symmetric travelling salesman problem. *Applied Soft Computing, 105*, Article 107298.

Park, J., Karumanchi, S., & Iagnemma, K. (2015). Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *IEEE Transactions on Robotics, 31*(5), 1101–1115.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems magazine, 22*(3), 52–67.

Patle, B. K., Pandey, A., Jagadeesh, A., et al. (2018). Path planning in uncertain environment by using firefly algorithm. *Defence technology, 14*(6), 691–701.

Patle, B. K., Ganesh, B. L., Pandey, A., et al. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology, 15*(04), 582–606.

Pei, W., Huayu, G., Zheqi, Z., et al. (2019). A novel hybrid firefly algorithm for global optimization. In *2019 IEEE 4th international conference on computer and communication systems (ICCCS)* (pp. 164–168). IEEE.

Phung, M. D., & Ha, Q. P. (2021). Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Applied Soft Computing, 107*, Article 107376.

Preparata F.P., Shamos M.I., Zhuang X.G., Introduction to computational geometry[J]. 1990.

Qi, J., Yang, H., & Sun, H. (2020). MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment. *IEEE Transactions on Industrial Electronics, 68*(8), 7244–7251.

Qian, K., Liu, Y., Tian, L., et al. (2020). Robot path planning optimization method based on heuristic multi-directional rapidly-exploring tree. *Computers & Electrical Engineering, 85*, Article 106688.

Qiuyun, T., Hongyan, S., Hengwei, G., et al. (2021). Improved Particle Swarm Optimization Algorithm for AGV Path Planning. *IEEE Access, 9*, 33522–33532.

Rahmaniar, W., & Rakhmania, A. E. (2022). Mobile Robot Path Planning in a Trajectory with Multiple Obstacles Using Genetic Algorithms. *Journal of Robotics and Control (JRC), 3*(1), 1–7.

Ravankar, A. A., Ravankar, A., Emaru, T., et al. (2020). HPPRM:Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots. *IEEE Access, 8*, 221743–221776.

Ryu, H., & Park, Y. (2019). Improved informed RRT* using gridmap skeletonization for mobile robot path planning. *International Journal of Precision Engineering and Manufacturing, 20*(11), 2033–2039.

Sadhu, A. K., Konar, A., Bhattacharjee, T., et al. (2018). Synergism of firefly algorithm and Q-learning for robot arm path planning. *Swarm and Evolutionary Computation, 43*, 50–68.

Samaniego, F., Sanchis, J., García-Nieto, S., et al. (2019). Recursive rewarding modified adaptive cell decomposition (RR-MACD): A dynamic path planning algorithm for UAVs. *Electronics, 8*(3), 306.

Sang, H., You, Y., Sun, X., et al. (2021). The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. *Ocean Engineering, 223*, Article 108709.

Sangeetha, V., Krishankumar, R., Ravichandran, K. S., et al. (2021a). Energy-efficient green ant colony optimization for path planning in dynamic 3D environments. *Soft Computing, 25*(6), 4749–4769.

Sangeetha, V., Krishankumar, R., Ravichandran, K. S., et al. (2021b). A fuzzy gain-based dynamic ant colony optimization for path planning in dynamic environments. *Symmetry, 13*(2), 280.

Santos, A. H. M., de Lima, R. M., Pereira, C. R. S., et al. (2019). Optimizing routing and tower spotting of electricity transmission lines: An integration of geographical data and engineering aspects into decision-making. *Electric Power Systems Research, 176*, Article 105953.

Šeda, M. (2007). Roadmap methods vs. cell decomposition in robot motion planning. In *Proceedings of the 6th WSEAS international conference on signal processing, robotics and automation* (pp. 127–132). Athens, Greece: World Scientific and Engineering Academy and Society (WSEAS).

Sepehri, A., & Moghaddam, A. M. (2021). A motion planning algorithm for redundant manipulators using rapidly exploring randomized trees and artificial potential fields. *IEEE Access, 9*, 26059–26070.

Shah, H., Tairan, N., Garg, H., et al. (2018). Global gbest guided-artificial bee colony algorithm for numerical function optimization. *Computers, 7*(4), 69.

Shao, X., Zheng, R., Wei, J., et al. (2021). Path planning of mobile Robot based on improved ant colony algorithm based on Honeycomb grid. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 1358–1362). IEEE.

K. Sharma S. Singh R. Doriya Optimized cuckoo search algorithm using tournament selection function for robot path planning[J] International Journal of Advanced Robotic Systems 18 3 2021 1729881421996136.

Shi, Q., Li, C., Wang, C., et al. (2017). Design and implementation of an omnidirectional vision system for robot perception. *Mechatronics, 41*, 58–66.

Shin, Y., & Kim, E. (2021). Hybrid path planning using positioning risk and artificial potential fields. *Aerospace Science and Technology, 112*, Article 106640.

Shivgan R., Dong Z., Energy-efficient drone coverage path planning using genetic algorithm. 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR). IEEE, 2020: 1-6.

Song, F. Q. (2021). Improved firefly algorithm and its application and research in vehicle routing problem[D]. *Jiangxi University of Finance and Economics*. https://doi.org/10.27175/d.cnki.gjxcu.2021.000640

Song, P. C., Pan, J. S., & Chu, S. C. (2020). A parallel compact cuckoo search algorithm for three-dimensional path planning. *Applied Soft Computing, 94*, Article 106443.

Souza, F. R., Câmara, T. R., Torres, V. F. N., et al. (2019). Mine fleet cost evaluation-Dijkstra's optimized path. *REM-International Engineering Journal, 72*, 321–328.

Stentz, A. (1997). *Optimal and efficient path planning for partially known environments [M]//Intelligent unmanned ground vehicles* (pp. 203–220). Boston, MA: Springer.

Stenzel, J., Lünsch, D., & Schmitz, L. (2021). Automated topology creation for global path planning of large AGV fleets. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 3373–3380). IEEE.

Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future generation computer systems, 16*(8), 889–914.

Sun, C. Y. (2021). Research on Trajectory Planning and Path Tracking Control Strategy for Intelligent Vehicle Emergency Collision Avoidance[D]. *Beijing Jiaotong University*. https://doi.org/10.26944/d.cnki.gbfju.2021.000262

Sun, H. (2017a). *Studies on Trajectory Planning Considering Motion Uncertainties of Traffic Vehicles[D]*. Jilin University.

Sun, L. (2017b). A real-time collision-free path planning of a rust removal robot using an improved neural network. *Journal of Shanghai Jiaotong University (Science), 22*(5), 633–640.

Su Z., Zhou X., Cheng T., et al., Global localization of a mobile robot using lidar and visual features. 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2017c: 2377-2383.

Szczepanski, R., & Tarczewski, T. (2021). Global path planning for mobile robot based on Artificial Bee Colony and Dijkstra's algorithms. In *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)* (pp. 724–730). IEEE.

Szczepanski, R., Tarczewski, T., & Erwinski, K. (2022). Energy Efficient Local Path Planning Algorithm Based on Predictive Artificial Potential Field. *IEEE Access, 10*, 39729–39742.

Tan, C. S., Mohd-Mokhtar, R., & Arshad, M. R. (2021). A Comprehensive Review of Coverage Path Planning in Robotics using Classical and Heuristic Algorithms. IEEE. *Access*.

Tang, H. W. (2019). Research on Multi-Robot Target Searching Based on Intelligent Optimization Algorithms in Unknown Environments[D]. *Hunan university*. https://doi.org/10.27135/d.cnki.ghudu.2019.004690

Tang, J., Chen, G., & Coon, J. P. (2018). Route selection based on connectivity-delay-trust in public safety networks. *IEEE Systems Journal, 13*(2), 1558–1567.

Tao, Y., Gao, H., Ren, F., et al. (2021). A Mobile Service Robot Global Path Planning Method Based on Ant Colony Optimization and Fuzzy Control. *Applied Sciences, 11*(8), 3605.
Tao, Y., Wen, Y., Gao, H., et al. (2022). A Path-Planning Method for Wall Surface Inspection Robot Based on Improved Genetic Algorithm. *Electronics, 11*(8), 1192.
Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence, 99*(1), 21–71.
Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM, 45*(3), 52–57.
Tsardoulias, E. G., Iliakopoulou, A., Kargakos, A., et al. (2016). A Review of Global Path Planning Methods for Occupancy Grid Maps Regardless of Obstacle Density. *Journal of Intelligent & Robotic Systems, 84*(1–4), 829–858.
Wang, D., Chen, S., Zhang, Y., et al. (2021a). Path planning of mobile robot in dynamic environment: Fuzzy artificial potential field and extensible neural network. *Artificial Life and Robotics, 26*(1), 129–139.
Wang, H., Wang, W., Sun, H., et al. (2016). Firefly algorithm with random attraction. *International Journal of Bio-Inspired Computation, 8*(1), 33–41.
Wang, H., Cui, Z., Sun, H., et al. (2017a). Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism. *Soft Computing, 21*(18), 5325–5339.
Wang, H., Zhou, X., Sun, H., et al. (2017b). Firefly algorithm with adaptive control parameters. *Soft computing, 21*(17), 5091–5102.
Wang, H., Li, G., Hou, J., et al. (2022a). A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm. *Electronics, 11*(3), 294.
Wang, N., Xu, H., Li, C., et al. (2021b). Hierarchical path planning of unmanned surface vehicles: A fuzzy artificial potential field approach. *International Journal of Fuzzy Systems, 23*(6), 1797–1808.
Wang, S. B. (2020). Research on Interaction-Aware Predictive Trajectory Planning for Autonomous Vehicle in Dynamic Environments[D]. *University of science and technology of China*. https://doi.org/10.27517/d.cnki.gzkju.2020.001752
Wang, W., Zhao, J., Li, Z., et al. (2021c). Smooth Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. *Journal of Robotics*.
Wang, X., Kou, X., Huang, J., et al. (2021d). A Collision Avoidance Method for Intelligent Ship Based on the Improved Bacterial Foraging Optimization Algorithm. *Journal of Robotics, 2021*.
Wang, X., Zhang, H., Liu, S., et al. (2022b). Path planning of scenic spots based on improved A* algorithm. *Scientific Reports, 12*(1), 1–7.
Wang Y.F., Cao X.H., Guo X., Warehouse AGV path planning method based on improved A* algorithm and system short-term state prediction[J/OL]. Computer Integrated Manufacturing Systems:1-22[2022c-03-20].
Wang, Z., Li, H., & Zhang, X. (2019). Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach. *Automation in Construction, 97*, 220–228.
Wang, Z., Peng, J., & Ding, S. (2022d). A Bio-inspired trajectory planning method for robotic manipulators based on improved bacteria foraging optimization algorithm and tau theory. *Mathematical Biosciences and Engineering, 19*(1), 643–662.
Wei K., Gao Y., Zhang W., et al., A modified Dijkstra's algorithm for solving the problem of finding the maximum load path. 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT). IEEE, 2019: 10-13.
B. Wu Research on obstacle avoidance of multi-agent based on bionic optimization algorithm[D].Taiyuan 2021 University of Science and Technology 10.27721/d.cnki.gyzjc.2021.000260.
Wu, B., Chi, X., Zhao, C., et al. (2022). Dynamic Path Planning for Forklift AGV Based on Smoothing A* and Improved DWA Hybrid Algorithm. *Sensors, 22*(18), 7079.
Wu, C., Fu, X., Pei, J., et al. (2021a). A Novel Sparrow Search Algorithm for the Traveling Salesman Problem. *IEEE Access, 9*, 153456–153471.
Wu, J., Song, C., Ma, J., et al. (2021b). Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles. *IEEE Transactions on Intelligent Transportation Systems*.
Wu, Y., Wu, S., & Hu, X. (2021c). Multi-constrained cooperative path planning of multiple drones for persistent surveillance in urban environments. *Complex & Intelligent Systems, 7*(3), 1633–1647.
Xie, Y. N. (2016). *The research for the mobile robot path planning algorithm[D]*. Xi 'an University of Architecture and Technology.
Xue, J., & Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. *Systems Science & Control Engineering, 8*(1), 22–34.
Xu J., Han F.Q., Liu Q.X., et al., Bioinformation Heuristic Genetic Algorithm for Solving TSP[J/OL]. Journal of System Simulation: 1-10[2021-09-22].
Xue, T., Li, L., Shuang, L., et al. (2021). Path planning of mobile robot based on improved ant colony algorithm for logistics. *Mathematical Biosciences and Engineering, 18*(4), 3034–3045.
Xu, W., Yang, Y., Yu, L. T., et al. (2022). A global path planning algorithm based on improved RRT*. *Control and Decision, 37*(04), 829–838. https://doi.org/10.13195/j.kzyjc.2020.1354
Yang C., Tang Y., Zhou L., et al., Complete coverage path planning based on bioinspired neural network and pedestrian location prediction. 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). IEEE, 2018: 528-533.
Yang F., Lee D.H., Keller J., et al., Graph-based topological exploration planning in large-scale 3d environments. 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 12730-12736.
Yang, G. F., & Cai, Y. G. (2021). A Survey of Hybrid Particle Swarm Optimization Algorithm for Solving Vehicle Routing Problem and Its Variants. *Automation & Information Engineering, 42*(02), 7–13.
Yang, L., Fu, L., Li, P., et al. (2022). An Effective Dynamic Path Planning Approach for Mobile Robots based on Ant Colony Fusion Dynamic Windows. *Machines, 10*(1), 50.
Yan S., Research on path planning of auv based on improved ant colony algorithm. 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, 2021: 121-124.
X.H. Yan Y.L. Zhu Z.C. Zhang et al. A Review of Bacterial Optimization and Its Applications[J] Journal of Zhengzhou University (Engineering Science) 39 05 2018 pp. 1–10+21 10.13705/j.issn.1671-6833.2018.05.010.
Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)* (pp. 210–214). IEEE.
Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.
Yeap, W. K., & Jefferies, M. E. (1999). Computing a representation of the local environment. *Artificial Intelligence, 107*(2), 265–301.
Yi, X., Zhu, A., Yang, S. X., et al. (2016). A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments. *IEEE transactions on cybernetics, 47*(4), 974–983.
Yu J., Ge X.X., Optimal route planning of parking lot based on Dijkstra algorithm. 2017 International Conference on Robots & Intelligent System (ICRIS). IEEE, 2017: 221-224.
Yu, L., Kong, D., Shao, X., et al. (2018). A path planning and navigation control system design for driverless electric bus. *IEEE Access, 6*, 53960–53975.
Yu, S., Zhu, S., & Zhou, X. (2016). An Improved Firefly Algorithm Based on Nonlinear Time-varying Step-size. *International Journal of Hybrid Information Technology, 9*(7), 397–410.
Yu, W., Liu, J., & Zhou, J. (2021). A Novel Sparrow Particle Swarm Algorithm (SPSA) for Unmanned Aerial Vehicle Path Planning. *Scientific Programming, 2021*.
Yun, X., & Tan, K. C. (1997). A wall-following method for escaping local minima in potential field based motion planning. In *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97* (pp. 421–426). IEEE.
Zadeth, L. A. (1965). Fuzzy sets. *Information and control, 8*(3), 338–353.
Zagradjanin, N., Rodic, A., Pamucar, D., et al. (2021). Cloud-based multi-robot path planning in complex and crowded environment using fuzzy logic and online learning. *Information Technology and Control, 50*(2), 357–374.
Zhang, C. (2013). Path tracking of a mobile robot using inertial measurement unit. *Control Theory & Applications, 30*(03), 398–403.
Zhang, H., Lin, W., & Chen, A. (2018). Path planning for the mobile robot: A review. *Symmetry, 10*(10), 450.
Zhang, H., Sun, C. L., Zheng, Z. J., et al. (2015). A modified dynamic window approach to obstacle avoidance combined with fuzzy logic[C]. International Symposium on Distributed Computing and Applications for Business. *Engineering and Science*, 523–526.
Zhang J., Singh S., Visual-lidar odometry and mapping: Low-drift, robust, and fast. 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 2174-2181.
Zhang, L., Zhang, Y., & Li, Y. (2020a). Mobile robot path planning based on improved localized particle swarm optimization. *IEEE Sensors Journal, 21*(5), 6962–6972.
Zhang, L., Wang, Y., Zhao, X., et al. (2021a). Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. *Journal of Mechanical Science and Technology, 35*(7), 3171–3181.
R.K. Zhang Research on the Automatic Overtaking Control Method Based onthe Improved Artificial Potential Field[D].Dalian 2021 University of Technology 10.26991/d.cnki.gdllu.2021.002495.
Zhang, S., Zhou, Y., Li, Z., et al. (2016). Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Advances in Engineering Software, 99*, 121–136.
Zhang, S. C., Pu, J. X., Si, Y. N., et al. (2020b). Survey on Application of Ant Colony Algorithm in Path Planning of Mobile Robot.Computer. *Engineering and Applications, 56*(08), 10–19.
Zhang, S., Pu, J., & Si, Y. (2021c). An adaptive improved ant colony system based on population information entropy for path planning of mobile robot. *IEEE Access, 9*, 24933–24945.
S.C. Zhang Z. Ming A VNF-aware Virtualization Layer Constructing Algorithm Based on Adjustable Hop Count[J] Journal of Zhengzhou University(Engineering Science) 2021, 42(05), pp. 32–36+55 10.13705/j.issn.1671-6833.2021.05.008.
Zhang, T. W., Xu, G. H., Zhan, X. S., et al. (2022a). A new hybrid algorithm for path planning of mobile robot. *The Journal of Supercomputing, 78*(3), 4158–4181.
Zhang, W., Zhang, S., Wu, F., et al. (2021e). Path planning of UAV based on improved adaptive grey wolf optimization algorithm. *IEEE Access, 9*, 89400–89411.
Zhang, X., Wang, J., Fang, Y., et al. (2018a). Multilevel humanlike motion planning for mobile robots in complex indoor environments. *IEEE Transactions on Automation Science and Engineering, 16*(3), 1244–1258.
Zhang, X. F., & Wang, X. Y. (2019). Comprehensive Review of Grey Wolf Optimization Algorithm. *COMPUTER SCIENCE, 46*(03), 30–38.
Y. Zhang Research on Task Execution-Oriented Environment Modeling Problem for Horae Service Robots[D].Shandong 2021 University 10.27272/d.cnki.gshdu.2021.000319.
Zhang, Z., Guo, Q., Chen, J., et al. (2018b). Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification. *IEEE Access, 6*, 26022–26035.
Zhang, Z., He, R., & Yang, K. (2022b). A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Advances in Manufacturing, 10*(1), 114–130.
Zhao, C. W. (2020). A Survey on Artificial Neural Networks. *Shanxi Electronic Technology, 03*, 94–96.
Zhao, Y., Liu, G. L., Tian, G. H., et al. (2017). A Survey of Visual SLAM Based on Deep Learning. *ROBOT, 39*(06), 889–896. https://doi.org/10.13973/j.cnki.robot.2017.0889

Zhao, Y., Liu, H., & Gao, K. (2021). An evacuation simulation method based on an improved artificial bee colony algorithm and a social force model. *Applied Intelligence, 51*(1), 100–123.

Zhenyang, X., & Wei, Y. (2022). Mobile robot path planning based on fusion of improved A* algorithm and adaptive DWA algorithm. *Journal of Physics: Conference Series IOP Publishing, 2330*(1), Article 012003.

Zheyi C., Bing X., AGV path planning based on improved artificial potential field method. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). IEEE, 2021: 32-37.

Zhou, X., Yu, X., & Peng, X. (2018). UAV collision avoidance based on varying cells strategy. *IEEE Transactions on Aerospace and Electronic Systems, 55*(4), 1743–1755.

Zhu, D., & Yang, S. X. (2021). Bio-inspired Neural Network-based Optimal Path Planning for UUVs under the Effect of Ocean Currents. *IEEE Transactions on Intelligent Vehicles*.

Zhu, Y., Zhu, L., Long, Y., et al. (2021). The Wolf pack assignment rule based on ant colony algorithm and the path planning of scout ants in complex raster diagram. *Journal of Physics: Conference Series IOP Publishing, 1732*(1), Article 012084.

Ziegler, J., Werling, M., & Schroder, J. (2008). Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In *2008 IEEE intelligent vehicles symposium* (pp. 787–791). IEEE.

Zou, S. X., Wang, P., & Han, X. (2019). Improved path planning algorithm based on PRM. *Modular Machine Tool & Automatic Manufacturing Technique, 1*, 1–3.

Zou, X. B. (2005). *Researeh on design of control system and environment modeling for a prototype of mobile robot*. Central South University.

Zuo X., Geneva P., Lee W., et al., Lic-fusion: Lidar-inertial-camera odometry. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 5848-5854.