

# Multi-Strategy Improved Petal Kingfisher Algorithm for Robot Path Planning

Guogang Wang, Hongwei Sun, Shouliang Xie, Xu Chen, Zichao Feng

Jilin Institute of Chemical Technology, Jilin, Jilin 132000, China

**Abstract:** Aiming at the problems of slow convergence speed and easy to fall into local optimal solutions of the original spotted kingfisher algorithm, this paper proposes an improved spotted kingfisher algorithm, which firstly initializes the population through Logistic chaotic mapping to improve the uniformity of the population position, secondly introduces the learning decay rate to dynamically adjust the search step to accelerate the convergence of the algorithm; and adopts Gaussian random variation strategy to avoid the algorithm from falling into the local optimum. The algorithm is Experiments are conducted on six kinds of benchmark test functions, and the results show that the improved algorithm significantly improves the original PKO algorithm in terms of solution accuracy and convergence speed, which effectively improves the performance of the algorithm. Finally, it is applied to robot path planning in Matlab and ROS, which shows that the improved algorithm has shorter path planning length, shorter convergence time and better robustness than other optimization algorithms.

**Keywords:** Mobile Robot; Path Planning; Optimization Algorithm; Logistic Chaos Mapping.

## 1. Introduction

With the rapid development of artificial intelligence and robotics, mobile robots have been used in many fields, such as intelligent manufacturing, warehousing and logistics, medical care, etc[1-2], path planning as the core link of autonomous navigation of mobile robots, the purpose is to allow the robot to move safely from the starting point to the end point in the known environment, and excellent path planning technology can accelerate the productivity of factories, reduce the degree of human labor, and improve the social and economic development play a key role. Excellent path planning technology can accelerate factory production efficiency, reduce the degree of human labor, and improve social and economic development play a key role. For this reason, many scholars at home and abroad have carried out relevant research on path planning technology, traditional path planning methods including RRT algorithm [3], A\* algorithm[4], Dijkstra algorithm [5] and PRM algorithm[6], although they can plan a safe path without collision. But in the face of complex environments, there are still limitations, for example, the path generated by the traditional RRT algorithm is not smooth and is not suitable for the robot to move quickly, and the A\* algorithm occupies a large amount of storage space in the face of complex scenarios, resulting in a waste of computational resources, based on these problems, heuristic functions have become a hotspot for scholars to study in recent years, which include CPO algorithm[7], DA algorithm[8], ESOA algorithm[9] and so on. These algorithms show strong global search and local development ability by simulating swarm behavior, and have strong adaptability to the environment.

Xu Guanghui et al[10]proposed an improved crown porcupine optimization algorithm, which improves the adaptability and robustness of UAVs in complex terrain by introducing inelastic collision principle and comprehensive cost function evaluation system; Chen et al [11] proposed an improved snowy egret optimization algorithm, which employs a synergistic combination of sit-and-wait and proactive strategies to enhance the exploration of the solution

space and the ability of local exploitation; Huang Huajuan et al [12] proposed an improved dragonfly optimization algorithm, which introduces Logistic chaos mapping in the population initialization stage to improve the initial solution diversity, and adopts cosine perturbation strategy to avoid the algorithm falling into local optimum. Although the above literature has improved the optimization algorithm and improved the global search ability of the algorithm to a certain extent, there is still much room for improvement in the convergence speed and convergence accuracy. In this paper, we propose an Improved Patterned Kingfisher Algorithm (IPKO), which uses logistic chaotic mapping instead of the traditional randomized initialization of the individual positions of the Patterned Kingfisher, to increase the diversity of the population, and make it more homogeneous. Secondly, the search step of the algorithm is dynamically adjusted by introducing the learning decay rate, which accelerates the convergence speed of the algorithm, and the introduction of Gaussian random variation strategy can improve the local search ability.

## 2. Traditional Kingfisher Algorithm

Pied Kingfisher Optimizer (PKO)[13]is a meta-heuristic optimization algorithm based on group intelligence, inspired by the hunting behavior and symbiotic relationship of Pied Kingfisher in nature. PKO algorithm achieves high efficiency by simulating the hunting strategy of Pied Kingfisher in different stages. The PKO algorithm is divided into three phases: the exploration phase, the exploitation phase and the local escape phase. Each phase simulates the behavior of the spotted kingfisher in the natural environment, so as to achieve the solution of the complex optimization problem.

### 2.1. Initialization Phase

The PKO algorithm initiates the search process by randomly generating an initial population with the following formula:

$$\begin{cases} X_{ij} = LB + (UB - LB) \times r_1 \\ i = 1, 2, \dots, N \quad \text{and} \quad j = 1, 2, \dots, Dim \end{cases} \quad (1)$$

Where,  $X_{ij}$  is the  $i$ th spotted kingfisher individual in the  $j$ th dimension,  $UB$ ,  $LB$  is the upper and lower bounds of the search space;  $r_1$  is a random number in the interval  $[0,1]$ ; , where  $i,j$  are positive integers of  $[1,N]$  and  $i \neq j, N$ ,  $N$  is the overall population size.  $dim$  is the dimension. Provide diversified initial solutions for the subsequent search and lay the foundation for global exploration.

$$\begin{cases} T = \left( \exp(1) - \exp\left(\frac{t-1}{Max\_Iter}\right)^{\frac{1}{BF}} \right) \times \cos(Crest\_angles) \\ Crest\_angles = 2\pi \times rand \end{cases} \quad (2)$$

Where,  $t$  is the current number of iterations,  $Max\_Iter$  is the maximum number of iterations,  $Crest\_angles$  is the decay rate of the control index, which is set to a value of 8, and  $x$  is the random variation of the angle of the simulated bird's crown feather.

### 2.2.2. Hovering Strategy

The spotted kingfisher flaps its wings at high frequencies and can remain stable in the air and hunt. The expression for the  $T$ -value in the hover strategy is:

$$\begin{cases} T = b \times \left( \frac{t}{Max\_Iter} \right)^{\frac{1}{B}} \\ b = rand \times \left( \frac{PKO\_Fitness(j)}{PKO\_Fitness(i)} \right) \end{cases} \quad (3)$$

$$\begin{cases} X_i(s+1) = X_{best}(s) + \Phi \times \left( \frac{PKO\_Fitness(i)}{F_{best}} \right) \times \eta \times |\mu_2 \times X_{best}(t) - X_i(t)| \\ \Phi = \frac{\mu_3}{\sqrt{\pi/2}} \\ \eta = 1 - \left( \frac{t}{Max\_Iter} \right) \end{cases} \quad (4)$$

Where,  $X_{best}$  is the current optimal individual position,  $F_{best}$  is the historical best fitness,  $\mu_2, \mu_3$  is a standard normally distributed random number, and  $\eta$  is a linear decreasing factor used to balance the development intensity. So the formula for updating the current position of the spotted kingfisher is

## 2.2. Exploration Phase

The spotted kingfisher expands its field of view by alternating between perching and hovering behaviors, corresponding to the exploration phase of the algorithm, which aims to enhance the global search capability. Static observation of prey during perching and dynamic adjustment of view angle during hovering.

### 2.2.1. Perching Strategy

The parameter  $T$  characterizes the environment perception dynamics and is calculated as:

Where,  $\alpha$  reflects the effect of individual fitness differences on the frequency of wing beats, and  $PKO\_Fitness(i)$  is the fitness value of the  $i$ th individual.

## 2.3. Exploitation Phase:

Simulation of the fast diving behavior of the spotted kingfisher to catch fish was used for the development phase, due to its ability to dive quickly and accurately and rapidly by virtue of its sharp beak and keen vision, which makes the spotted kingfisher an excellent aquatic hunter, the positional updating equation is:

$$\begin{cases} X_i(t+1) = X_i(t) + \alpha \times T \times (X_j(t) - X_i(t)) \\ \alpha = 2\mu_1 - 1 \end{cases} \quad (5)$$

Where,  $\mu_1$  is a standard normally distributed random number. This parameter is used to drive the individual to move towards the region with better adaptation.

## 2.4. Symbiosis Stage

Simulate the symbiotic relationship between the spotted

$$\begin{cases} X_i(t+1) = \begin{cases} X_m(t) + o \times \alpha \times |X_i(t) - X_n(t)| & rand > (1-\eta) \\ X_i(t) & otherwise \end{cases} \\ \theta = \theta_{\max} - (\theta_{\max} - \theta_{\min}) \times \frac{t}{Max\_Iter} \end{cases} \quad (6)$$

Where,  $X_m(t), X_n(t)$  is the position of two randomly selected individuals,  $\theta$  is used to dynamically adjust the escape probability, and  $\theta_{\max}, \theta_{\min}$  is set to 0.5, 0.  $o$  is the hunting ability parameter, which is calculated by adaptation.

## 3. Improvement of the Flamboyant Kingfisher Algorithm

### 3.1. Population Initialization by Invoking Logistic Chaotic Mapping

The traditional PKO uses uniform random distribution to generate the initial solution, which may lead to uneven coverage of the search space. Introducing Logistic chaotic mapping to generate the initial population, the sequence generated by Logistic mapping has traversal and mutual dissimilarity, which can effectively avoid the initial solution aggregation with its mathematical expression as:

$$X_{k+1} = \zeta \times X_k (1 - X_k) \quad (7)$$

Where,  $\zeta$  is a control parameter, which generally takes the value of [3.57,4]. In this paper, the value is taken as 4. It is in a completely chaotic state.

### 3.2. Dynamic Adjustment of Parameters for Nonlinear Learning Decay Rate

Beating Factor (BF) in PKO algorithm controls the development intensity in the search phase, but its linear adjustment strategy is difficult to adapt to the complex search process. In this paper, we propose a hybrid learning rate model based on exponential decay and cosine annealing, which realizes the dynamic balance between global search and local exploitation by accelerating the initial exploration through the exponential term and smoothing the late convergence through the cosine term. The expression is as follows:

$$\phi(t) = \phi_0 \times \exp\left(-\rho \times \frac{t}{T}\right) + \nu \times \cos\left(\frac{\pi t}{2T}\right) \quad (8)$$

where,  $\phi_0$  is the initial learning rate,  $\rho$  is the decay coefficient,  $\nu$  is the annealing amplitude, and  $T$  is the total number of iterations.

### 3.3. Local Escape Mechanism of Gaussian Random Variation Strategy

Gaussian variants are introduced in the symbiosis phase to enhance the ability of the algorithm to jump out of the local optimum, and this strategy focuses on global perturbations at the beginning of the iteration, and gradually converges to fine search at the later stage.

kingfisher and other organisms such as otters, which is used for local escape to avoid falling into the local optimal solution.

$$\begin{cases} X_{mut} = X_{elite} + \zeta \times (U - L) \times N(0,1) \\ \zeta = 0.1 \times \left(1 - \frac{t}{T}\right) \end{cases} \quad (9)$$

Where,  $X_{elite}$  is the elite individual,  $\zeta$  is the adaptive standard deviation, and U and L are the search space boundaries.

### 3.4. Multi-strategy Improvement of the Flamboyant Kingfisher Optimization Algorithm Flow

In this paper, we propose to use multiple strategies to improve the spotted kingfisher optimization algorithm (PKO). Firstly, logistic chaos mapping is used to increase the initial diversity of the population; secondly, the learning decay rate is introduced to dynamically adjust the search step of the algorithm to accelerate the convergence speed; finally, Gaussian stochastic variation is used to avoid the algorithm from falling into a local optimum. The following are the specific steps of the multi-strategy improvement of the flower-spotted kingfisher optimization algorithm:

Step 1: Initialize the parameters of PKO algorithm, including the basic parameters such as the number of populations N, the dimensions of the claim space D, the maximum number of iterations x and so on;

Step 2: Initialize the spotted kingfisher population using logist chaotic mapping, calculate and rank the population fitness, and find the current global optimal solution x ;

Step 3: In the search phase, the individual kingfisher adjusts its search step according to Eq. (8) to update its position;

Step 4: In the exploitation phase, operate Eq. (9) on the current spotted kingfisher population, recalculate individual fitness, and keep the mutated individuals if the fitness improves, otherwise keep the original individuals;

Step 5: Judge whether the algorithm reaches the termination condition, output the optimal solution if the termination condition is reached, otherwise repeat steps 2-step 4.

## 4. Experimental Results and Analysis

### 4.1. Experimental Parameter Settings

Selected spotted kingfisher optimization algorithm (PKO), crown porcupine optimization algorithm (CPO), dragonfly optimization algorithm (DA), egret swarm optimization algorithm (WSOA), multi-strategy spotted kingfisher optimization algorithm (IPKO), this paper's experimental environment: processor AMD Ryzen 77435H ,graphics card

RTX4060, the main frequency of 3100 MHz, memory 16 GB, the operating system is 64-bit Windows11 computer, using MATLAB R2024a, all algorithms were run independently for 50 times, the maximum population size of 30, the maximum number of times, the experimental parameters were set up. The operating system is 64-bit Windows11 on the computer, using MATLAB R2024a, all the algorithms are run independently 50 times, the population size is 30, and the maximum number of iterations is set to 500. 6 commonly used benchmark test functions are selected for the simulation experiments as shown in Figure 1, and the test functions are shown in Table 1.

## 4.2. Analysis of Experimental Results

From Figure 1, it can be seen that the improved algorithm in this paper in the single-peak function, multi-peak function and composite function under the convergence speed and are

faster than other algorithms, in the single-peak function test, IPKO algorithm convergence speed is faster than the other optimization algorithms, can be in the early iteration to quickly find the location of the optimal solution, in the multi-peak function test, the IPKO algorithm has a stronger convergence accuracy and the ability to jump out of the local optimal solution, composite function In the multi-peak function test, IPKO algorithm has stronger convergence accuracy and ability to jump out of the local optimal solution, and the convergence curve of IPKO algorithm is significantly higher than PKO algorithm in the middle and late stages of the test, which indicates that IPKO algorithm is better than other optimization algorithms in terms of convergence speed, convergence accuracy, and optimization searching ability, and it is more suitable for the ability of complex optimization problems and better robustness.

Table 1. Benchmark function specific information

function type	Function name	function code	dimension	variable interval	minimum value
single-peak function	Bent Cigar	F1	30	[100,100]	0
	Zakharov	F2	30	[-10,10]	0
multifunction	Griewank	F3	30	[-1.28,1.28]	0
	Ackley	F4	30	[-5.12,5.12]	0
composite function	Pathological	F5	3	[-600,600]	0
	Penholder	F6	4	[-32,32]	0

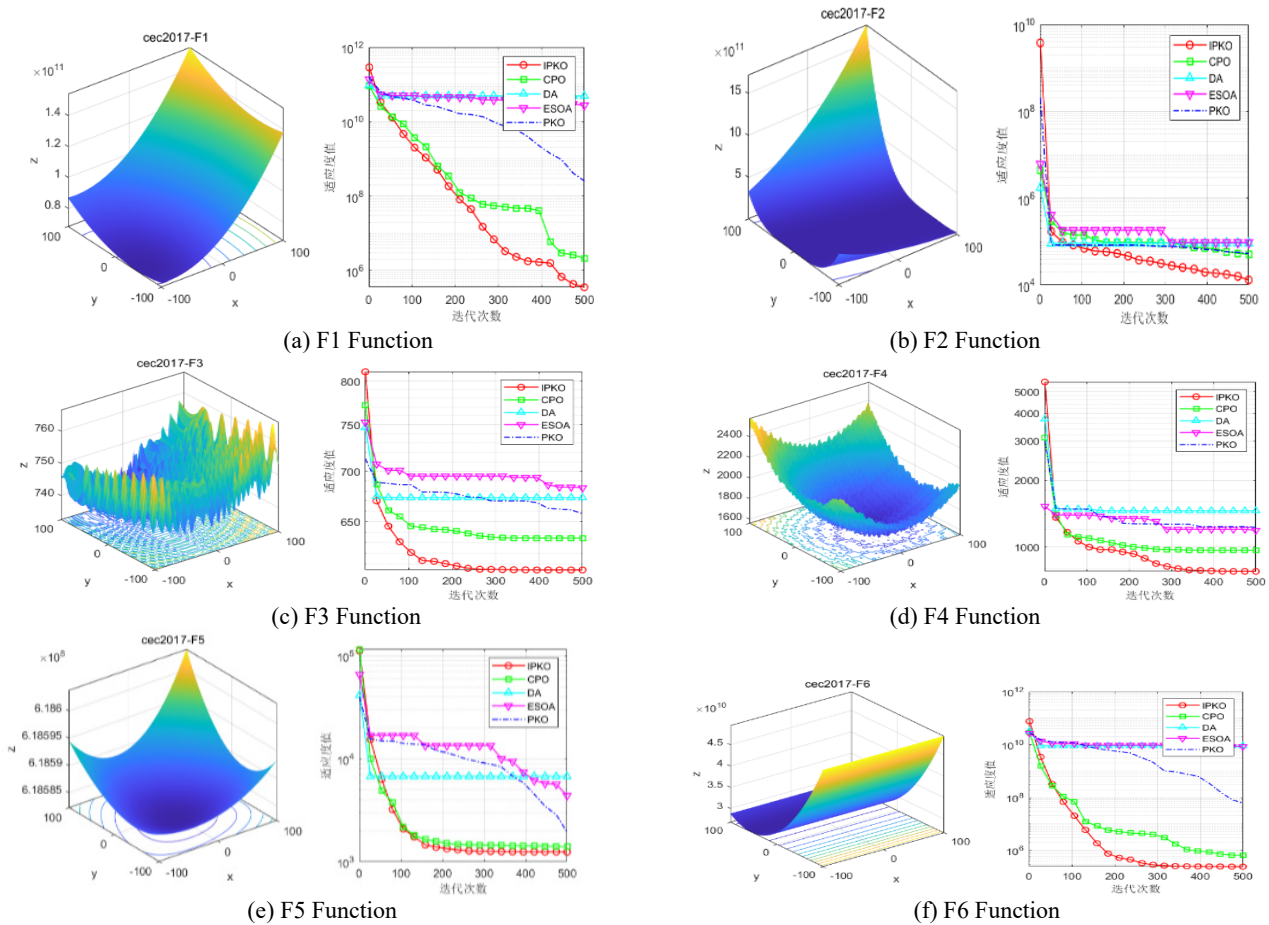


Fig 1. Convergence curves of each algorithm under the benchmark test function

## 5. Robot Path Planning Simulation Verification

### 5.1. Path Planning Experiment

In order to verify the application of the spotted kingfisher

optimization algorithm in mobile robot path planning, this paper selects 20 20 simple raster map environment and 50 50 complex raster map environment. At the same time calculated the number of turns in the path planning algorithm, path length and convergence time as the key indicators of the

performance of each algorithm, these indicators not only take into account the speed of the algorithm to reach the optimal position, but also measures the degree of optimization of the algorithm and the practicality of the planning, in order to

ensure that the algorithm in the path planning of the fairness, the maximum number of iterations are set to 500, the number of populations are 30.

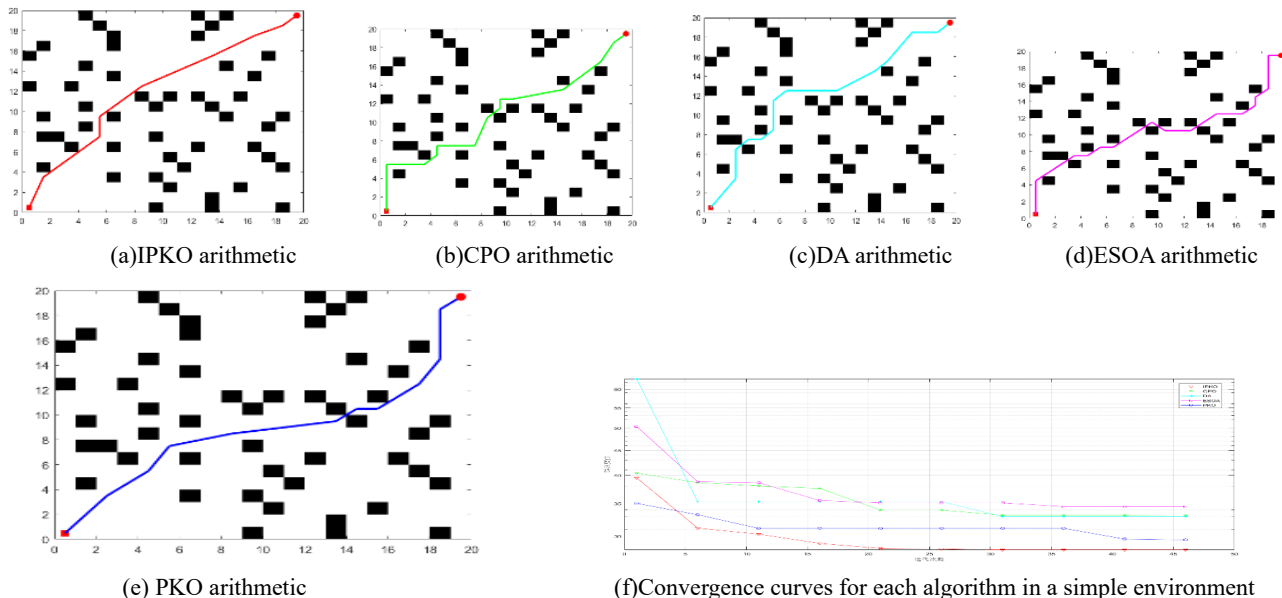


Fig 2. Experimental diagram of mobile robot path planning in simple environment

Table 2. Experimental data of mobile robot path planning in simple environment

in the end	IPKO	PKO	DA	CPO	ESOA
Path length	28.1486	29.8243	30.8877	32.0067	32.9706
Number of turns	7	10	12	12	14
Average convergence time/second	0.2589	0.562	0.5809	0.6813	0.9777

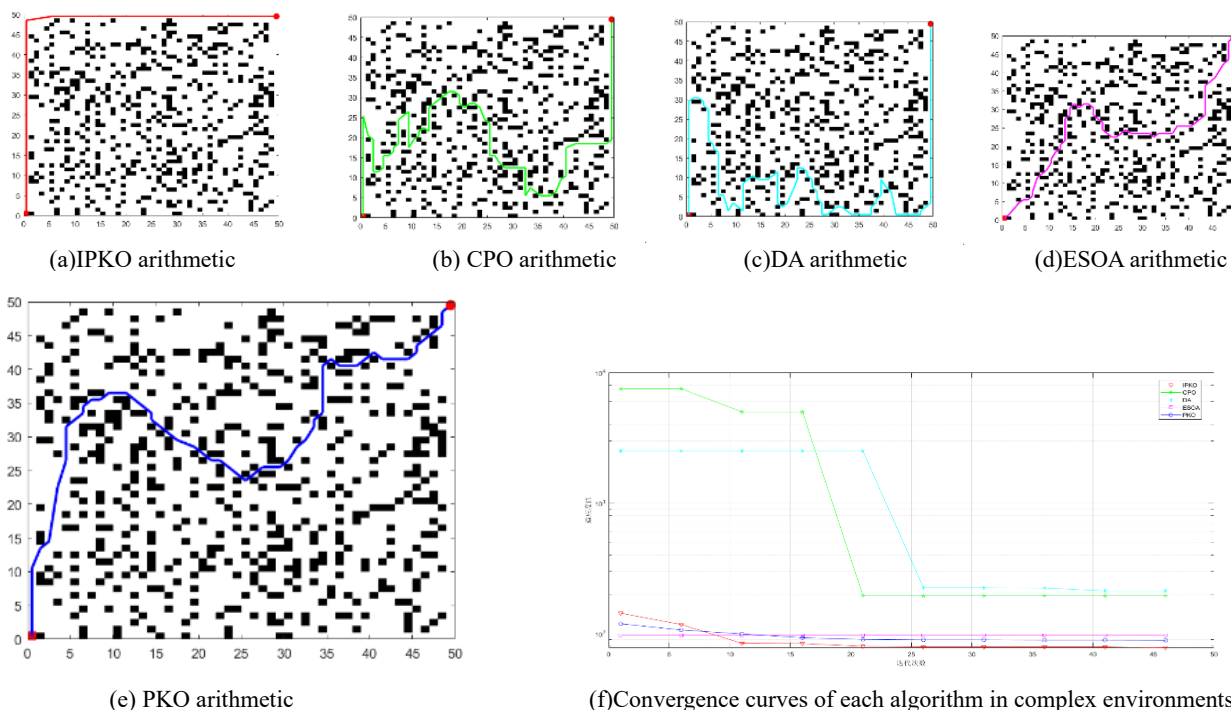


Fig 3. Experimental map of mobile robot path planning in complex setting

Table 3. Comparison of various algorithms in complex map setting

in the end	IPKO	PKO	DA	CPO	ESOA
Path length	97.099	104.8965	105.8182	184.309	205.1863
Number of turns	2	38	41	46	50
Average convergence time/second	0.4377	0.562	2.7676	0.0758	0.0359

## 5.2. Simulation Experiment Analysis

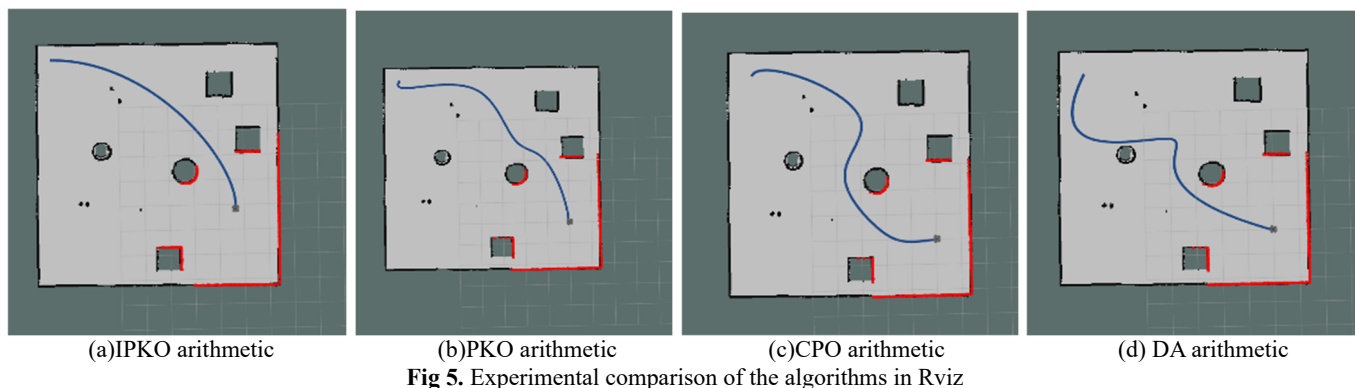
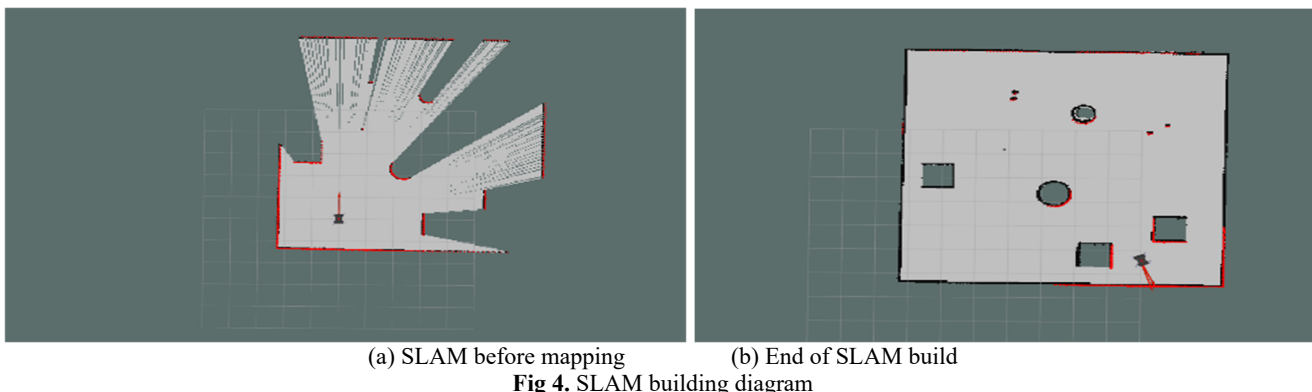
From the above graphs, it can be seen that IPKO algorithm outperforms other optimization algorithms in terms of convergence time, and in specific analysis, the complex setting is shortened by 53.93%, 55.43%, 61.99%, and 73.52% compared to PKO, DA, CPO, and ESOA, respectively. In the simple setting, compared with PKO, DA, CPO and ESOA, the shortening is 22.12%, 84.19%, 82.68% and 91.79% respectively, which proves that the improved algorithm can reach the target point quickly in a short period of time, and it can be seen that the improved algorithm makes fewer turns, has better smoothing, and is more suitable for the robot's movement, and in terms of the path length, compared with PKO, DA, CPO and ESOA, the shortening is 53.93%, 55.43%, 61.99% and 73.99% respectively in the simple setting. In terms of path length, it is shortened by 5.62%, 8.87%, 12.05% and 14.62% in simple environment compared with PKO, DA, CPO and ESOA respectively. In the complex setting, it reduces 7.43%, 8.24%, 47.32% and 52.68% compared with PKO, DA, CPO and ESOA, respectively. The improved

algorithm has significant advantages in path planning, and the above analysis proves that the improved algorithm has higher search efficiency, smoother trajectory and shorter planning time in path planning.

## 6. ROS Simulation Verification Robot Path Planning Experiment

### 6.1. ROS Map Building and Autonomous Navigation Simulation

In this paper, we use Ubuntu 22.04 ROS1 Noetic version to build a real environment in Gazebo experimental platform, set up human-shaped obstacles and multiple rectangular obstacles, use Gmapping algorithm to complete the SLAM construction as shown in Fig 4. Next, we verify the performance of the four algorithms in path planning in RVIZ. All set the same starting point and end point, each algorithm for 30 times path planning, the path planning of the four optimization algorithms is shown in Figure 5.



**Table 4.** Experimental results of the four algorithms in RVIZ

in the end	IPKO	CPO	DA	ESOA	PKO
Path length/m	14.25	18.95	27.14	29.99	38.41
Average convergence time/second	10.62	14.98	23.83	27.95	32.69

As can be seen from Table 4, this paper's algorithm still shows strong performance in the real setting, and can complete the planning in a short period of time with a shorter path length, and at the same time, the paths planned by this paper's algorithm have a high degree of smoothness, which is more suitable for the robot to move in the complex scene, proving that the algorithm's stability and planning efficiency are both higher.

## 7. Conclusion

In order to overcome the problems encountered by the original PKO algorithm in robot path planning, such as slow convergence speed, low search efficiency and low trajectory smoothness, an improved spotted kingfisher algorithm is proposed, which firstly enriches the diversity and expansion of the population by initializing the population through logist chaotic mapping. To avoid premature convergence of the

spotted kingfisher algorithm, secondly, through the nonlinear learning decay rate and Gaussian random variation strategy, it can improve the convergence accuracy and global search ability, and finally applied in the robot path planning, the simulation experimental results have shown that the improved algorithm has a stronger stability, faster convergence speed and excellent trajectory smoothness in the path planning. The performance of the improved algorithm exceeds that of the original spotted kingfisher algorithm and other optimization algorithms.

## References

- [1] Ravi R, Andrzej K .A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives [J].Applied Sciences,2022,12(14):6951-6951.
- [2] LI Zhonglin, LUO Shaoping,JIA Yuting. A review of path planning algorithms for mobile robots[J]. Modern Information Technology,2024,8(19):184-188.
- [3] Liu Shanshan, Jin Ming. Mobile robot path planning algorithm based on improved RRT[J/OL]. Journal of Qufu Normal University (Natural Science Edition),1-7[2025-06-02].
- [4] Gao Huanchao. Research on path planning of picking robot based on A\* algorithm[J]. Research on Agricultural Mechanization, 2025,47(07):240-244.
- [5] Xianming S, Haobang L ,Yuan L , et al.Location Planning of Field Ammunition Depot for Multi-stage Supply Based on Dijkstra Algorithm[J].Journal of Physics: Conference Series, 2021, 2068(1).
- [6] AviramS ,LevnerE .Enhancing Path Planning for Autonomous Robots in Large, Obstacle-Crowded Environments: A Practical Improvement to the PRM Algorithm[J].Journal of Robotics, 2025, 2025(1):9569965-9569965.
- [7] Wang Xinwei, Feng Feng. Research on two-dimensional path planning for robots based on crown porcupine optimization algorithm [J/OL]. Internet of Things Technology,1-4[2025-06-02].
- [8] SONG Qingjun, MENG Xiangfu, JIANG Haiyan,et al. Mobile robot path planning based on improved dragonfly algorithm [J/OL]. Combined machine tools and automated machining technology,1-7[2025-06-02].
- [9] ZHENGJUN ZHAO,LIKUN HU,CHENGJIE CAI, et. al. Mobile robot path planning based on improved egret swarm optimization algorithm[J]. Modern Manufacturing Engineering, 2025, (02):37-43+150.
- [10] Guanghui Xu, Eddie Deng, Shuqing Wang,et al. Improved Crested Porcupine Optimization Algorithm for Three-Dimensional Path Planning of Unmanned Aerial Vehicles[J]. Journal of Central South University for Nationalities (Natural Science Edition),2025,44(02):237-244.
- [11] Zuyan C, Adam F ,Shuai L , et al.Egret Swarm Optimization Algorithm: An Evolutionary Computation Approach for Model Free Optimization.[J].Biomimetics (Basel, Switzerland), 2022, 7 (4):144-144.
- [12] HUANG Huajuan, MIN Feng. A multi-strategy dragonfly algorithm for solving inverse kinematics[J]. Journal of Henan Normal University (Natural Science Edition),2023,51(05):46-58.
- [13] Sulaiman H M, Mustaffa Z ,Saari M M , et al.Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems[J].Engineering Applications of Artificial Intelligence,2020,87103330-103330.