IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 September 2023, accepted 1 October 2023, date of publication 10 October 2023, date of current version 16 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323437

RESEARCH ARTICLE

Robot Path Planning Based on Interval Type-2 Fuzzy Controller Optimized by an Improved Aquila Optimization Algorithm

KUN LI, XIANG ZHANG, AND YING HAN
Faculty of Electrical and Control Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China

Corresponding author: Ying Han (hyfengyan@163.com)

This work was supported in part by the Liaoning Revitalization Talents Program under Grant XLYC2007091, and in part by the National Natural Science Foundation of China under Grant 62203197.

ABSTRACT Uncertainty and complexity in the local path planning are hot topics. In this paper, a novel IAOFC algorithm is proposed for local path planning in the complex environment. Considering the uncertainty and complexity of local path planning, this paper uses interval type-2 fuzzy control to design path planning method, which can respond more quickly to the uncertainty of the environment and improve the computation speed and efficiency. To further improve the performance of the fuzzy controller, this paper uses an improved Aquila Optimizer (AO) algorithm to optimize the membership function of the interval type-2 fuzzy controller (renamed by IAOFC). By using the optimized fuzzy controller, the time cost and path cost can be reduced. In simulation experiments, the path planning in static environment is designed to verify the basic performance and efficiency of the algorithm, and the path planning in dynamic environment is validated to verify the robustness of the algorithm. Finally, the superiority of the IAOFC algorithm is proved by comparing it with some other algorithms. According to experiment results, IAOFC has an average cost reduction of 15% and 6% than other algorithms in static and dynamic environments, respectively.

INDEX TERMS Path planning, interval type-2 fuzzy control, Aquila optimization algorithm, membership function, uncertainty.

I. INTRODUCTION

In recent years, robots have been widely used due to their excellent intelligence and flexibility [1]. Especially in the harsh working environment (mine, nuclear reactor, disaster rescue, etc.), the use of intelligent robots can effectively reduce the work risk so that the operator can complete the work task safely and efficiently. In most scenarios, autonomous navigation is a necessary condition for intelligent mobile robots, of which path planning is one of the most important parts of navigation [2].

Path planning has always been a hot topic in robotics research, such as the classical algorithms based on graph theory: A* [3], Dijkstra [4], etc. Sampling based algorithms: rapidly exploring random tree(RRT) [5], Probabilistic roadmaps(PRM) [6], etc.; Neural network based algorithm: bionic neural network [7], residual convolutional neural network [8], particle swarm optimization neural network [9], dynamic environment path planning neural network [10], potential field bio-inspired neural network [11], etc. There are also popular biological inspirations in recent years, such as: genetic algorithm(GA) [12], particle swarm optimization(PSO) [13], ant colony optimization(ACO) [14], etc. However, they all rely on the prior environmental information, and sometimes it is difficult for the robot to obtain the prior environmental information in the actual work process, such as the mine disaster relief site. The change of the geographical environment after the disaster and the urgency of the rescue make it insufficient to realize the robot's autonomous path planning under the prior environment.

Artificial potential field(APF) [15] can perform path planning in unknown environment, but it has the local minimum problem, which may not find the solution to the problem. Dynamic window approach(DWA) [16] has low

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

computational complexity and can perform real-time path planning, but it has poor obstacle avoidance effect and may not be able to find the optimal path. Due to the restriction of objective conditions, there are many disturbances and uncertainties that cannot be eliminated in mobile robot path planning, such as inaccurate sensor information, errors in robot implementation, dynamically changing environment, and so on. Therefore, it is necessary to choose the right method to deal with these problems. Using the concept of fuzzy membership function, the fuzzy logic algorithm discriminates fuzzy sets and simulates the behavior of human brain to deal with fuzzy relations. It has strong adaptability and can control and adjust the robot in real time under different environments to better adapt to the complex and changing external environment. It can deal with the non-linear relationship between input and output [22], and has good robustness to noise, disturbance and parameter changes, so as to ensure the stability and reliability of the robot's action, and meanwhile the fuzzy control can deal with the non-linear relationship between input and output. In this regard, the robot path planning and design method based on fuzzy control is preferentially selected in this paper.

Nowadays, the fuzzy logic method has been widely studied and applied, such as: medical supply chain [17], solar dryer [18], aircraft [19], high order multi-agent system [20], pattern recognition [21]. Song et al. [23] proposed a path planning algorithm based on fuzzy logic under unknown environment, which defines the direction of advance and uses fuzzy reasoning to get the priority of each direction; then, the direction of the robot is the direction with the highest priority; However, this algorithm only considers the direction of the robot, without its speed. Singh et al. [24] proposed a type-1 fuzzy logic controller design method, but it cannot perform well in the face of high complexity environment. Compared with type-1 fuzzy control, interval type-2 fuzzy control uses interval number to describe the relationship between input variables and output variables. It has better fault tolerance, accuracy and interpretability, and has better performance in solving some complex problems, such as Microgrid frequency regulation [25], local model control [26], three-phase PWM rectifier control [27], Type-2 fuzzy C-means algorithm [28]. Dirik et al. [29] used interval type-2 fuzzy logic for the path planning, but the algorithm is based on known environment. In this paper, considering unknown environment for the path planning problem, the optimized interval type-2 fuzzy controller is designed, during to the fact that some uncertain parameters make the performance of the fuzzy controller does not reach the optimum. Therefore, the swarm intelligence optimization algorithm is employed to optimize the interval type-2 fuzzy controller. Aquila Optimizer (AO) [30] is a swarm intelligence optimization algorithm with excellent global optimization ability, fast convergence speed, etc. It has advantages to quickly find the optimal parameter of the membership function when there are many fuzzy rules in the fuzzy controller, which can better save optimization time. In order to better increase the optimization performance of the fuzzy controller for the path planning problem, a new Improved Aquila Optimizer (IAO) is proposed in this paper. To solve the problem of easily falling into the local optimum, the spiral foraging strategy of the Whale Optimization Algorithm (WOA) and the danger warning mechanism of the Sparrow Search Algorithm (SSA) are introduced. At the same time, in order to better jump out of the local region, this paper uses fusion Cauchy and opposition-based learning strategies to disturb the optimal position after each iteration. The IAO algorithm is applied to the Interval type-2 fuzzy controller, and the relevant parameters of the shape of the membership function are optimized. The optimized Interval type-2 fuzzy controller is then used for path planning in static and dynamic environments.

Our major contributions are summarized as follows:

(1) An improved IAO optimization algorithm is proposed. The whale spiral foraging strategy and the hazard warning mechanism are integrated to increase the optimization performance. On this basis, the opposition-based learning fused by the Cauchy strategy is added to improve the ability of jumping out of the local optimum original algorithm;

(2) An optimized interval type-2 fuzzy controller by IAO is proposed. The parameter setting of the size and shape of the control membership function is regarded as a multidimensional optimization problem to optimize the membership function;

(3) A path planning method based on interval type-2 fuzzy control under unknown environment is proposed, which reduces the dependency on the previous environment, makes more adaptable to the realistic complex environment, and increases the robustness to deal with uncertain environment.

The remaining of this paper is arranged as follows. Section II introduces the robot motion model, the type-2 fuzzy system and the path planning method by the type-2 fuzzy controller. An IAO algorithm is proposed in Section III, which improves the optimization performance in combination with some useful mechanisms. Section IV describes a robot path planning method based on optimized interval type-2 fuzzy controller. In Section V, the algorithm is tested in different simulation experiments and compared with other algorithms. Section VI summarizes the thesis and discusses the future work.

II. PROBLEM DESCRIPTION

In this paper, a path planning method based on interval type-2 fuzzy control is proposed for path planning, positioning and motion control problems in mobile robot control. The method extracts visual information from the robot sensor and uses IT2FIS to generate a path plan to reach the desired target position.

A. ROBOT MODEL

When moving, the robot has visual sensors in three directions, namely the front distance detector, the left front distance detector and the right front distance detector, as shown in Figure.1.

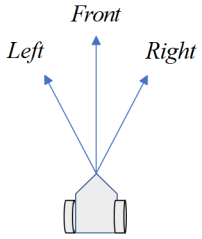

FIGURE 1. Schematic diagram of robot sensors.

The kinematic equation of the robot is shown in the following equation.

$$\dot{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u \tag{1}$$

$$u = [v\ \omega]^T \tag{2}$$

$$q = [x, y, \theta]^T \tag{3}$$

where q is the current pose state of the robot; $[x, y]$ represents the position of the robot under the current coordinates; θ is the angle between the robot's current direction of motion and the X axis; v is the linear velocity of the robot; ω is the angular velocity of the robot rotating around its own center.

B. OBSTACLE AVOIDANCE BASED ON VISION

Robot path planning is performed in an unknown environment. The robot moves from the starting point, recognizes the environment and avoids obstacles at the same time, as shown in Figure. 2. The θ_r is the angle between the robot's moving direction and the horizontal direction, the θ shows the angle between the robot's position and the target position, and the θ_e is the angle between the robot's direction and the target direction.

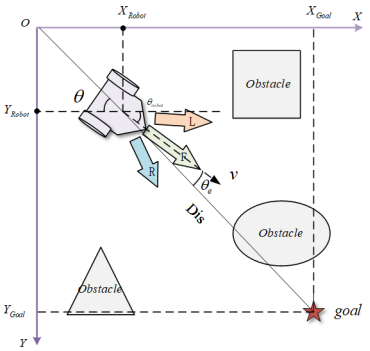

FIGURE 2. Schematic diagram of robot path planning.

The distance between the robot and the obstacle is showed as follows:

$$D_{\text{obstacle-robot}} = \min(DLF, D\,RF, D\,F); \tag{4}$$

where $D_{obstacle-robot}$ represents the distance between the robot and the obstacle, DLF, DRF, DF are respectively the distance between the left front of the robot, the right front of the robot and the obstacle of the front.

The orientation of the mobile robot in the global coordinate system can be described by the central point of its position and the angle. Assuming that the initial position and the target position of the robot are available, the position error is calculated as follows:

$$e_x = X_t - X_r = \text{Dis} * \cos(\theta_r) \tag{5}$$

$$e_y = Y_t - Y_r = \text{Dis} * \sin(\theta_r) \tag{6}$$

where Dis represents the distance between the mobile robot and the target, (Xt, Yt) and (Xr, Yr) respectively represent the position of goal and robot, the calculation of Dis is as follows:

$$\text{Dis} = \sqrt{(e_x)^2 + (e_y)^2} \tag{7}$$

The robot heading direction is calculated as follows:

$$\theta_r = \tan^{-1}\frac{e_y}{e_x} \tag{8}$$

The angle between the robot's forward direction and the target direction is calculated as follows:

$$\theta_{\text{e}} = \theta_{\text{r}} - \theta \tag{9}$$

The decision to turn is made by combining the angle difference with the distance to the obstacle. Setting the Turn variable will always make the robot more oriented towards the target point. If the obstacle ahead is close and the same as the obstacle in the previous step, the turning operation in the previous step is repeated.

In the obstacle avoidance based on unknown environment vision, the information of the surrounding obstacles is collected by the vision sensors in the three directions of left front, front and right front, and the closest obstacle in the three directions is selected as the current distance between the robot and the obstacle. According to the distance between the robot and the obstacle, the robot's speed is updated as follows:

$$\theta_{\text{e}} = \theta_{\text{r}} - \theta \tag{10}$$

The direction difference between the robot and the target position can be obtained from equations (5)-(9), and variables such as DLF, DRF, DF, Turn, Dis can be mapped to the fuzzy domain. The angular velocity of the robot output in the next step can be obtained by fuzzy reasoning according to the established fuzzy database. The velocity of the robot can be obtained by combining equation (10), and path planning can be performed in the unknown environment.

C. PATH PLANNING BASED ON INTERVAL TYPE-2 FUZZY CONTROLLER

In 1975, L. A. Zadeh proposed type-2 fuzzy sets [32]. An interval type-2 fuzzy set can be represented by the following formula in [33]:

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} \mu\tilde{A}(x,u)/(x,u),\ where\ J_x \in [0,1] \tag{11}$$

$$\tilde{A} = \int_{x\in X}\left[\int_{u\in J_x} 1/u\right]/x,\ where\ J_x \in [0,1] \tag{12}$$

Type-2 fuzzy systems use an FOU field to represent the uncertainty in [32], which is a bounded function consisting of two type-1 member functions. The FOU of $\tilde{A}$ can be expressed as:

$$FOU(\tilde{A}) = \bigcup_{\forall x \in X} J_x = \{(x, u) : u \in J_x \subseteq [0, 1]\} \quad (13)$$

where J_x represents the union of x and u that can be taken.

The upper function can be expressed as:

$$\bar{\mu}_{\tilde{A}}(x) = FOU(\tilde{A}) \ \ \forall x \in X \quad (14)$$

The lower function can be expressed as:

$$\underline{\mu}_{\tilde{A}}(x) = FOU(\tilde{A}) \forall x \in X \quad (15)$$

Membership function of type-2 fuzzy system is shown in Figure. 3.

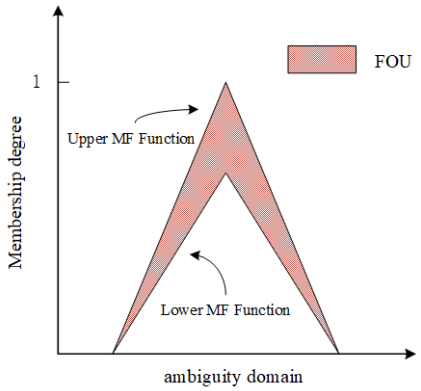

FIGURE 3. Interval type two fuzzy membership function.

The type-2 fuzzy control system includes fuzzy module, rule base module, fuzzy reasoning module and output processing module, as shown in Figure. 4. Different from type-1 fuzzy control system, type-2 fuzzy control system also has a type reductionist before output, which can convert type-2 fuzzy output into type-1 fuzzy output.

In the fuzzy control algorithm of this paper, we choose four fuzzy variables as the input variables of the robot: angle difference to the target (θ_e), distance to the target (Dis), distance to the obstacle (DL, DF, DR), avoid the obstacle and turn (Turn). The output variable is the angle of the robot (Steer). The overall structure of the fuzzy control logic is shown in Figure. 5.

The angle to the target is the difference between the robot's own travel angle and the robot's position and the angle of the target. The robot will turn by this angle to face the target. The range of values for this variable is [−180,180]. The distance to the target is normalized to the interval [0,1]. The closest distance between the robot and the obstacle is taken as the distance between the robot and the obstacle in three directions at the same time (left front, front, right front) and this variable is mapped to [0,1]. The algorithm also calculates whether the robot will turn left or right based on the angle to the obstacle. Four input variables are fed into the reasoning system and

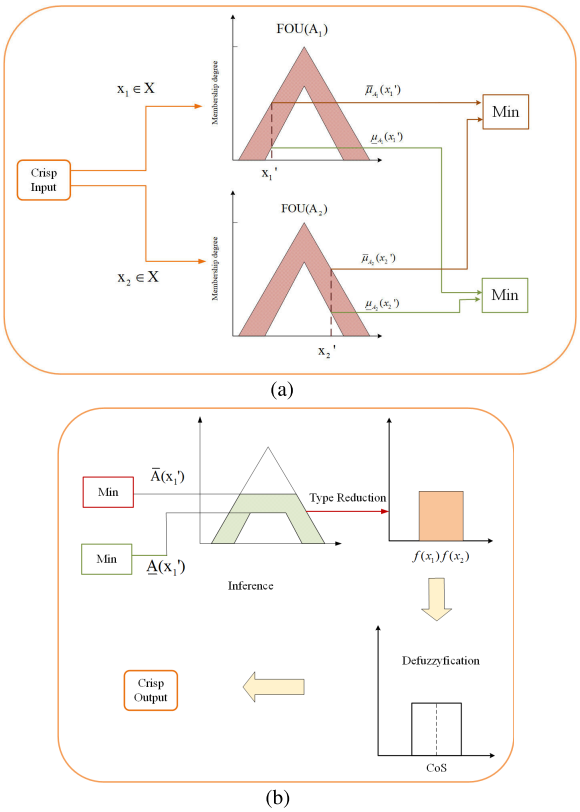

FIGURE 4. The architecture of interval type-2 fuzzy logic control.

the angle of the robot's turn to avoid the obstacle is obtained. The speed of the robot is determined by the distance from the obstacle. Compared with other membership functions, the Gaussian membership function has smoothness, which makes it able to adapt to the asymptotic relationship between fuzzy variables. In addition, the Gaussian membership function is highly tunable. The relevant parameters of the membership function are shown in Table 1.

Membership functions for input and output variables are shown in Figure.6.

In order to describe the relationship between input variables and output variables, guide the operation of the control system, and successfully complete the task of path planning, we use IF-THEN rules to construct database of fuzzy rules. Suppose there are input variables M, N, output variables are represented by R, and M points belong to three different fuzzy subsets A_1, A_2, A_3, and N points belong to three different fuzzy subsets B_1, B_2, B_3, and R belong to C_1, C_2, C_3. The relationship between M, N, R can be described by the following rule:

IF M is A_1 and N is B_2, THEN R is C_3;

In this paper, 25 rules [29] are set according to IF-THEN rules to control the robot's motion mode, and the rule base is shown in Table 2. DLF, DRF, and DF respectively represent the distance between obstacles in front of the robot on the left, right, and front. The first four rules make basic judgments based on the distance between the robot and the

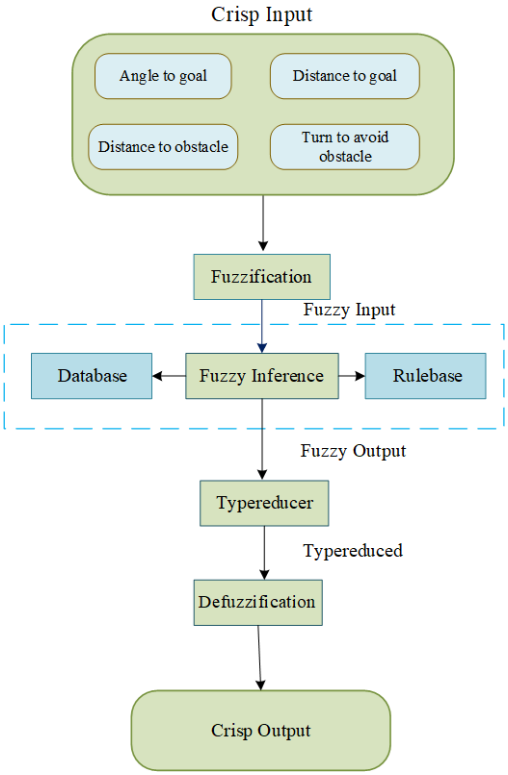

FIGURE 5. Flow chart of interval Type-2 Fuzzy control path planning algorithm.

TABLE 1. Parameters of membership function.

Fuzzy Variable	Fuzzy Subset	Membership Function	Function Parameters
DistanceFront	Low	Gaussmf	[0.03635 6.94e-18],[1],[0.2 0.2]
	Medium	Gaussmf	[0.0641 0.194],[1],[0.2 0.2]
	Large	Sigmf	[24.9 0.3694],[1],[0.2 0.2]
DistanceFrontLeft	Small	Gaussmf	[0.028 6.94e-18],[1],[0.2 0.2]
	Medium	Gaussmf	[0.043 0.14822],[1],[0.2 0.2]
DistanceFrontRight	Small	Gaussmf	[0.028 6.94e-18],[1],[0.2 0.2]
	Medium	Gaussmf	[0.043 0.1482],[1],[0.2 0.2]
Anglegoal	Negative	Gaussmf	[0.0766 -0.224],[1],[0.2 0.2]
	No	Gaussmf	[0.0441 1.39e-17],[1],[0.2 0.2]
	Positive	Gaussmf	[0.0766 0.2501],[1],[0.2 0.2]
	MoreNegative	Sigmf	[-26 -0.3921],[1],[0.2 0.2]
	MorePositive	Sigmf	[28.8 0.395],[1],[0.2 0.2]
Turn	Left	Trimf	[-1.8 -1 -0.795],[1],[0.2 0.2]
	Right	Trimf	[0.595 1 1.8],[1],[0.2 0.2]
DistanceGoal	Low	Gaussmf	[0.036 6.94e-18],[1],[0.2 0.2]
	Low	Gaussmf	[0.036 6.94e-18],[1],[0.2 0.2]
	Medium	Gaussmf	[0.064 0.194],[1],[0.2 0.2]
	High	Sigmf	[24.9 0.3694],[1],[0.2 0.2]
Steer	Left	Gaussmf	[0.0821 -0.24782],[1],[0.2 0.2]
	No	Gaussmf	[0.0692 0],[1],[0.2 0.2]
	Right	Gaussmf	[0.0857 0.260],[1],[0.2 0.2]
	MoreLeft	Gaussmf	[0.135 -0.608],[1],[0.2 0.2]
	MoreRight	Gaussmf	[0.134 0.646],[1],[0.2 0.2]

obstacle. If there is an obstacle on the left, it turns right. The Turn variable is a factor that comprehensively considers the angle between the robot's forward direction and the endpoint direction, as well as the distance from the obstacle in front, to determine the turning direction. The goal of the rule is to make the robot more oriented towards the target. The remaining rules are composed of fuzzy subsets of AG

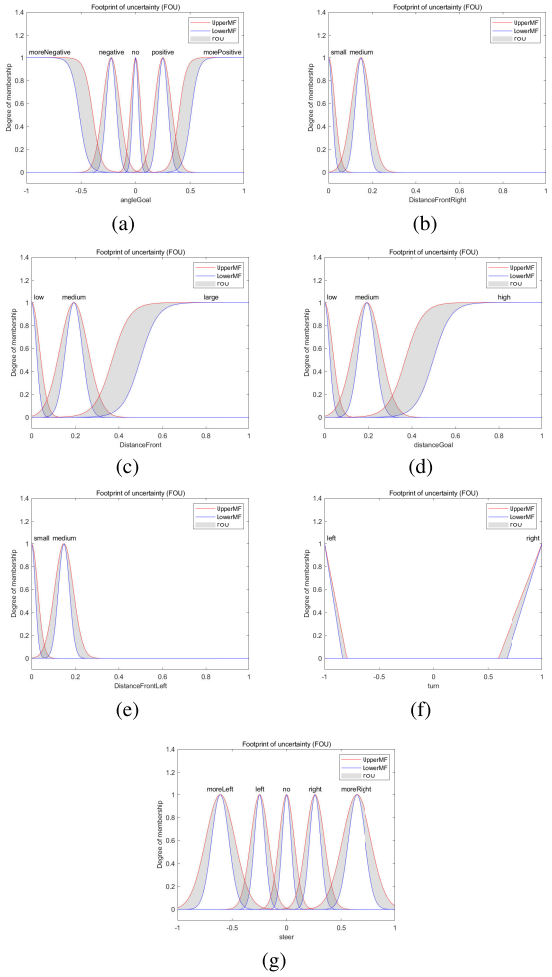

FIGURE 6. Membership function of input variables(a)-(f) and output variables(g).

and DG, where AG represents the angle between the robot's forward direction and the endpoint, and DG represents the distance between the robot and the endpoint. Consider the combination of AG and DG subsets in different situations to better guide the robot to reach the endpoint.

TABLE 2. Path planning rules by Type-2 fuzzy control.

	DLF	DRF	DF	Turn	AG	DG	Steer
1	Small						More Right
2	Medium						Right
3		Small					More Left
4		Medium					Left
5			Low	Left			More Left
6			Medium	Left			Left
7			Large	Left			Left
8			Low	Right			More Right
9			Medium	Right			Right
10			Large	Right			Right
11					Negative	Low	More Left
12					More Negative	Low	More Left
13					No	Low	No
14					Positive	Low	More Right
15					More Positive	Low	More Right
16					Negative	Medium	Left
17					More Negative	Medium	No
18					No	Medium	Right
129					Positive	Medium	More Right
20					More Positive	High	Left
21					Negative	High	Left
22					More Negative	High	No
23					No	High	Right
24					Positive	High	Right
25					More Positive	Medium	More Left

The robot will receive the information directly from the external environment to the interval type-2 fuzzy system, through the fuzzy reasoning to generate the robot's steering information, and calculate the robot speed by the distance between it and the obstacle, so that the robot can find an optimal route to the destination in a relatively safe and stable state path.

In the interval type-2 fuzzy control, there is no need for fine modelling, the system is robust, and the control instability caused by modelling error, noise and other unstable factors in the traditional method is avoided. However, at the same time, the parameters of the fuzzy membership function depend on subjective sets, which makes it cannot reach the optimal setting, and the performance of the fuzzy controller will be limited. So in this paper, an improved Aquila optimizer is used to optimize the parameters of the membership function of the fuzzy controller.

III. IMPROVED AQUILA OPTIMIZER

A. CLASSICAL AQUILA OPTIMIZER

1) EXPANDED EXPLORATION

Aquila identifies prey areas and selects the best hunting areas by hovering in the sky, then it performs a vertical dive behavior in the air in [30] to determine the area of the search space where the prey is located. The mathematical model of this behavior is as follows:

$$X_1(t+1) = X_{best}(t) \times (1 - \frac{t}{T}) + (X_M(t) - X_{best}(t) * rand) \tag{16}$$

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), \forall j = 1, 2 \cdots, Dim \tag{17}$$

where $X_1(t+1)$ is the solution generated by the next iteration of the extended exploration phase; $X_{best}(t)$ is the optimal solution, reflecting the approximate position of the prey; $X_M(t)$ represents the mean position of the current solution connected at the t iteration; *rand* is the random number between (0,1); t and T represent the current and maximum iterations, respectively; *Dim* is the dimension; N is population size.

2) SMALL-RANGE EXPLORATION

In this stage, the Aquila hovers constantly over the prey after finding it at a high altitude, preparing to land and then launch an attack. This behavior is called the short glide attack in [30]. The mathematical description of this behavior is as follows:

$$\begin{cases} X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (y - x) * rand \\ Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \\ \sigma = (\frac{\Gamma(1+\beta) \times \sin e(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}) \\ y = r \times \cos(\theta) \\ x = r \times \sin(\theta) \\ r = r_1 + U \times D_1 + \theta_1 \\ \theta = -\omega \times D_1 + \theta_1 \\ \theta_1 = \frac{3 \times \pi}{2} \end{cases} \tag{18}$$

where $X_2(t+1)$ is the solution generated by the next iteration of the small-range exploration stage; D is the dimension; $Levy(D)$ is levy flight distribution function; $X_R(t)$ is the random solution taken in the range $[1, N]$ at the i^{th} iteration; s is a fixed constant of 0.01, u and v is a random number between 0 and 1; β is a fixed value of 1.5; y and x represent spiral shapes in the search; r_1 take the value from [1,20]; U is a constant with the value 0.00565; D_1 is integer numbers from 1 to the length of the search space; ω is a fixed value of 0.005.

3) EXPANDED DEVELOPMENT

In this phase, when the Aquila accurately determines the prey area and prepares to land and attack, this mode is called low altitude flight with slow descent in [30]. The mathematical description of this behavior is

$$\begin{aligned} X_3(t+1) = (X_{best}(t) - X_M(t)) \times \alpha - rand \\ + ((UB - LB) \times rand + LB) \times \delta \end{aligned} \tag{19}$$

where $X_3(t+1)$ is the solution produced by the next iteration of the expanded development phase; $X_{best}(t)$ represents the position of the optimal solution until the i^{th} iteration; $X_M(t)$ represents the mean value of the current solution at the t iteration; *rand* is a random value between 0 and 1; α and δ is two development tuning parameters fixed to 0.1 in this paper. LB represents the lower bound and UB represents the upper bound.

4) SMALL-RANGE DEVELOPMENT

In this phase, Aquila attacks prey on the land according to their own random movements in [30]. The Aquila will attack the prey in the last position and the mathematical description of this behavior is

$$\begin{cases} X_4(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times rand) \\ -G_2 \times Levy(D) + rand \times G_1 \\ QF(t) = t^{(1-T)^2} \\ G_1 = 2 \times rand - 1 \\ G_2 = 2 \times (1 - \frac{t}{T}) \end{cases} \tag{20}$$

where $X_4(t+1)$ is the solution for the next iteration of the small-range development phase t; QF represents a quality function for balancing the search strategy; G_1 represents the various movements of tracking its prey; G_2 is a decreasing value from 2 to 0, indicating the speed at which a aquila follows its prey; $X(t)$ is the current solution at the t iteration.

B. CLASSICAL AQUILA OPTIMIZER

1) OPPOSITION-BASED LEARNING WITH CAUCHY MUTATION

It can be seen from the previous analysis that the mechanism of the update strategy in the AO makes it difficult to escape if it becomes locally trapped. Therefore, it is very important to perturb the current optimal position again after the iteration to make it possible to jump out of the local optimum. In this paper, we use a opposition-based learning strategy that fused by Cauchy mutation. The target position is perturbed and

updated to prevent the algorithm from falling into the local optimum.

The opposition-based learning was proposed by Tizhoosh in [34], whose purpose was to find the corresponding opposition solution based on the current solution, and select and store the better solution through evaluation. In order to better guide the individual to find the optimal solution, the opposition-based learning is integrated into AO, and the mathematical description is

$$\begin{cases} X^*_{best}(t) = ub + r \oplus (lb - X_{best}) \\ X_{New}(t+1) = b_3 \oplus (X_{best}(t) - X^*_{best}(t)) \end{cases} \tag{21}$$

where $X^*_{best}(t)$ is the opposition solution of the target solution at t iteration; $X_{New}(t+1)$ is the target solution at t+1 iteration; ub and lb are upper and lower bounds; r is a random number matrix of 1ÃUdim uniformly distributed by (0, 1) criteria (where dim is the dimension of the search space); b_3 is the pseudo-information exchange coefficient, expressed by

$$b_3 = (\frac{T_{\max} - t}{T_{\max}})^t \tag{22}$$

Bring the Cauchy operator into the target position update. The Cauchy operator has a long step length, and the longer distribution at both ends can make the individual have a higher probability to jump to a better position and escape from the local optimum. Meanwhile, the peak value of the smaller center point indicates that the time spent on searching the domain space of the Cauchy operator is less, and improving the ability of jumping out of the local optimum. The mathematical description in [41] is

$$X_{New}(t+1) = Cauchy \oplus X_{best}(t) \tag{23}$$

To improve the optimization performance of the algorithm, the opposition-based learning strategy and the Cauchy operator perturbation strategy are alternately executed with a certain probability, and the target position is dynamically updated randomly. In the opposition-based learning strategy, the opposition solution is obtained by the general opposition-based learning strategy, and the search range of the algorithm is increased. Meanwhile, ub and lb of the upper and lower bounds in equation (29) change dynamically. In the Cauchy mutation strategy, mutation operators are used to mutate the optimal position to generate new solutions, which ameliorates to some extent the defect that the algorithm easily falls into the local optima. The selection probability for deciding which strategy to choose for updating is defined by

$$P_s = -\exp(1 - \frac{t}{T_{\max}})^{20} + \theta \tag{24}$$

The selection probability of deciding which strategy to choose for updating is defined as follows:

If rand < P_s
  Update target position according to equation 21.
else
  Update target position according to equation 23
If the fitness is better, update the optimal location
end

2) SPIRAL FORAGING STRATEGY

The exploitation phase of the Aquila Optimizer is in the late stage of the iterations, which will lead to the gradual assimilation of the population, resulting in the algorithm falling into the local optimum. To solve this, the spiral foraging strategy in the WOA is introduced in this paper. In the process of hunting, the whale shrinks the net while spirally encircling prey in [35]. The behavior of the spiral foraging is defined by

$$\begin{cases} X_{New}(t+1) = D \cdot e^{bl} \cos(2\pi l) + X^*(t) \\ D = |X^*(t) - X(t)| \end{cases} \tag{25}$$

where b is a constant and l is a random number located at [−1,1].

According to equation 25, for the position update, the aquila can imitate the whale to conduct the spiral foraging in the local area, which can strengthen the local exploration ability and the search ability for blind spots, and increase the coverage of the search space.

3) DANGER WARNING MECHANISM

The danger warning mechanism in [36] is employed to enhance ability to jump out of the local optimum. When individuals at the edge of a sparrow flock perceive danger, they will quickly move to a safe position. This behavior is described as follows:

$$X_{New}(t+1) = \begin{cases} X^t_{best} + \beta |X(t) - X^t_{best}| & if\ f_i > f_g \\ X(t) + K \cdot (\frac{|X(t) - X^t_{worst}|}{(f_i - f_w) + \varepsilon}) & if\ f_i = f_g \end{cases} \tag{26}$$

Suppose that X_{best} is the current global optimal position; β is the step size control parameter; f_g is the current global optimal fitness value; K is a random number in [−1,1]; f_w is the fitness values of the current individuals; f_i is the current global worst fitness value; ε is the smallest constant which is used to avoid the denominator being 0. When $f_i > f_g$, it indicates that the sparrow is at the edge of the group; X_{best} represents the center position of the population, and it is safe around it; $f_i = f_g$ indicates that the sparrow in the middle of the population is aware of the danger and needs to get close to other sparrows;

There are many random factors in the classical algorithm, and although the use of Levy flight can jump out of the local area, it also reduces the ability of the algorithm to dig deeply for the local area, and it is easy to miss the global optimal value. In this paper, a part of aquilas are selected as the alarmer, when the marginal individual realizes the danger in its position, it can move to the safe position in time to avoid the attack of predators or other adverse effects. In this way, the errors and mistakes in the search can be reduced, and the search efficiency can be improved. At the same time, the communication among individuals within the population of aquilas can be strengthened, the dependence on the global optimal solution can be reduced, and the robustness and local search ability can be improved.

4) MAIN STEPS OF IAO

The improved Aquila optimizer in this paper combines the spiral foraging strategy of WOA and the danger warning mechanism of SSA, and uses the opposition-based learning strategy based on Cauchy to disturb the optimal position to jump out of the local optimum after each update, which makes the algorithm more flexible and able to find the optimal solution faster and more accurately. The flowchart of IAO is shown in Figure.7.

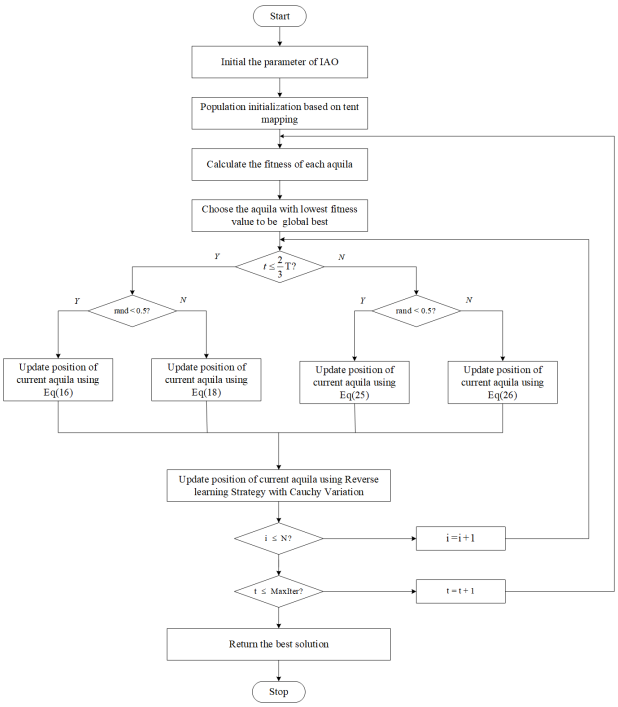

FIGURE 7. Flow chart of IAO.

The main steps of the IAO algorithm is shown in Algorithm 1.

5) COMPLEXITY ANALYSIS

Computational complexity of AO usually depends on the initialization of the solution, the calculation of fitness function value and the update of the solution in [34]. Let the population is N, O(N) is the population initialization calculation, $O(T \times N) + O(T \times N \times D)$ is the computational complexity of the update the solution,Where T is the number of iteration and D is dimension. Therefore, the total Computational complexity of AO is $O(N \times (T \times D + 1))$.

The spiral foraging strategy added in this paper replace the original model, so it will not increase the time complexity. Suppose that the time cost used to execute the opposition-based learning strategy with Cauchy variation is ω, then the computational complexity of the opposition-based learning strategy with Cauchy variation is as follows:

$$T_1 = O(\omega \times T \times N \times D) = O(T \times N \times D) \tag{27}$$

Algorithm 1 IAO

Input: *maxgen*, *popsize*, *bound_L*, *bound_H*
Output: the optimal X.

initialization, generate initial population:
for $t = 1, 2, \ldots,$ *popsize* **do**
    by formula $X_i = pos\_L + (bound\_H - bound\_L) \cdot rand()$;
end for
Choose the agent with lowest fitness value to be global best;
if $kreachesmaxgen/3$ **then**
    if $rand < 0.5$ **then**
        Update position of agent using equation 16;
    else
        Update position of agent using equation 24;
    end if
    Choose better solution;
else
    if $rand < 0.5$ **then**
        Update position of agent using equation 29
    else
        Update position of agent using equation 31
        Choose better solution;
    end if
end if
Update position of agent using Reverse Learning Strategy with Cauchy Variation;
Choose better solution;
Choose the agent with lowest fitness value to be global best;
Return best solution X;

Therefore, the total computational complexity of the IAO algorithm is as follows:

$$T = O(N \times (T \times D + 1)) + T_1 = O(N \times (T \times D + 1)) \tag{28}$$

In conclusion, the computational complexity of IAO is the same as that of standard AO.

IV. PATH PLANNING BASED ON OPTIMIZED INTERVAL TYPE-2 FUZZY CONTROLLER

In this paper, a robot path planning method (IAOFC) based on an optimized interval type-2 fuzzy controller is proposed. The general technique route of the algorithm is shown in Figure. 8.

The path planning fuzzy controller is preliminarily designed according to the obstacle avoidance principle of the robot in the unknown environment, and a simple interval type-2 fuzzy controller is obtained. The input variables of the controller are the distance between the obstacle (DLF, DRF, DF), the turn of the robot (Turn), the angle between the robot and the target (AG), the distance between the robot and the target (DG), the output of the controller is Steer.

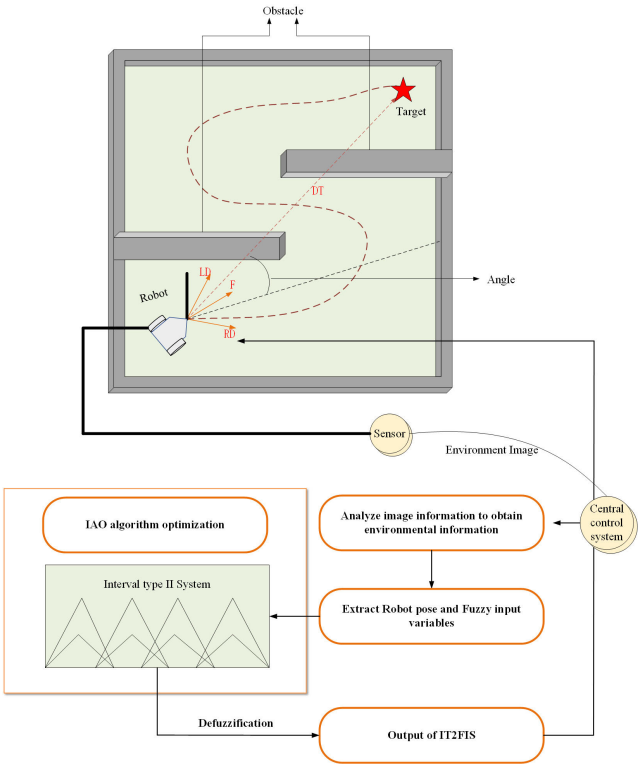

FIGURE 8. IAOFC algorithm technology roadmap.

In order to further improve the performance of the fuzzy controller, this paper uses the improved aquila optimizer to optimize the fuzzy controller. The parameter setting that controlling the shape of the membership function of each fuzzy variable is regarded as a high-dimensional optimization problem, and various parameters of UpperMF, LowerMF and LowerScale are optimized and written into the fuzzy system.

In the process of path planning, the robot collects information about the surrounding obstacles through the vision sensor and maps the physical information to the corresponding fuzzy domain and inputs it into the fuzzy controller. The fuzzy controller optimized by IAO is used for fuzzy reasoning to obtain the corresponding output fuzzy variable. At this point, the output is transformed into a type-1 fuzzy set by pattern reduction processing. Finally, the physical variables of controlling robot can be obtained by defuzzification. Flow chart of the IAOFC algorithm is shown in Figure.9.

V. SIMULATION EXPERIMENTS

A. PERFORMANCE TEST FOR IAO ALGORITHM

1) TEST FUNCTION

To verify the effectiveness of IAO, 23 test functions were used for optimization experiments in this paper in [36], including single-mode reference function (F1-F7), multi-mode reference function (F8-F16), and fixed-dimension multi-mode reference function (F17-F23). These test functions are listed in in Table 3. The experimental environment is: AMD Ryzen 7 4800H 2.9GHz CPU, 16GB memory, Windows 10 (64-bit), MATLAB 2019b.

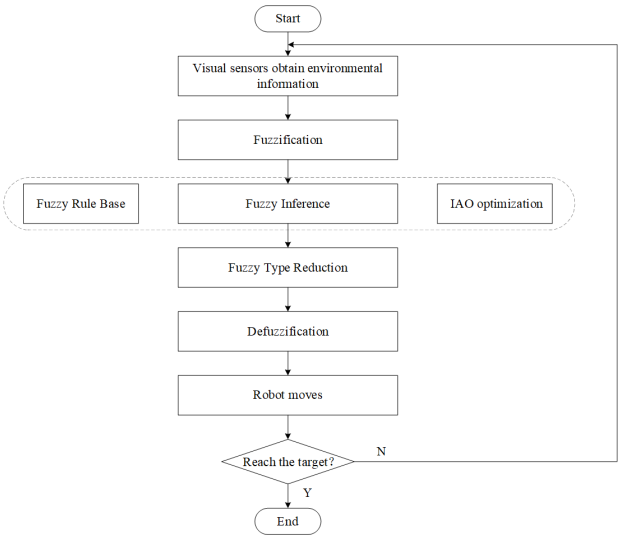

FIGURE 9. Flow chart of IAOFC algorithm.

2) EVALUATION INDEX

Mean value and standard deviation are used to evaluate the optimization performance of different algorithms, respectively defined as follows:

$$Mean = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{29}$$

$$Std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(S_i - Mean)^2} \tag{30}$$

where n is the number of iterations and S_i is the final result of each optimization.

In the simulation experiment, the proposed IAO algorithm is compared with other algorithms, including AO in [30], PSOGWO in [37], EOSMICOA in [39], LSSA in [38] and FASSA in [41]. The common parameters of all algorithms are set the same, the maximum iteration number is 500, the population number is 30, and running under the same environment. In general, the unimodal function is used to examine the local search ability of an algorithm, the multimodal function examines its global search ability, and the mixed and compound function examines its ability to deal with complex problems. In this paper, Mean and Std. are used to evaluate the optimization performance and stability of an algorithm, respectively. When solving minimum problems, a smaller Mean means better optimization performance, and a smaller Std. means a better stability.

3) PERFORMANCE ANALYSIS OF ALGORITHMS

In the unimodal test function, the convergence speed of the algorithm is more meaningful for the optimization results. This index reflects the ability of the algorithm to quickly

TABLE 3. Test functions.

Function	Range
$F_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[-100,100]
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[-100,100]
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	[-30,30]
$F_6(x) = \sum_{i=1}^{n} \left([x_i + 0.5]\right)^2$	[-100,100]
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	[-1.28,1.28]
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	[-500,500]
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[-5.12,5.12]
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]
$F_{12}(x) = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	[-50,50]
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50,50]
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	[-65,65]
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_i(b_i^2 + b_i x_x)}{b_i^2 + b_i x_3 + x_4}\right]^2$	[-5,5]
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5,5]
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	[-5,5]
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2$ $(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2$ $(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2,2]
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	[1,3]
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]
$F_{21}(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	[0,10]
$F_{22}(x) = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	[0,10]
$F_{23}(x) = -\sum_{i=1}^{10}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	[0,10]

find the global optimum region in the exploration phase. The optimization convergence curves of various algorithms on unimodal functions (F1-F7) are shown in Figure. 10.

It can be seen from Figure. 10 that IAO has excellent search and convergence speed on the unimodal function, and compared with AO, PSOGWO and EOSMICOA, IAO has stronger global search ability, although IAO is slightly slower than LSSA and FASSA in convergence speed. However, in F1, F2, F3, F4, F5 and F7 functions, the convergence position of IAO is closer to zero than that of LSSA and FASSA. Combined with the data marked in bold in Table 4, it can be seen that the Mean and Std. value of IAO are the smallest in 15 experiments, which indicates that IAO has maintained high stability in multiple experiments.

For the multimodal function, the optimization algorithm needs to get rid of much local minimum interference and finally find the global optimal value, so the test on the multimodal function is more to test the local optimization ability of the algorithm. The convergence curves of different algorithms on multimodal functions (F8-F16) are shown in Figure. 11.

The local search ability of AO is poor, and it can search quickly in the global scope, but it often enters the convergence in advance without better development of the local area, resulting in missing the global optimal value. The IAO has made various improvements on the defect that the original AO is prone to fall into the local optimality. As shown in Figure. 11(a), Figure. 11(e), Figure. 11(f), and Figure. 11(h), it can be seen that the ability of IAO to overcome falling into the local optimality has been significantly improved compared to AO. For multimodal functions, the number of local minima increases exponentially with increasing dimensions. According to the results of Table 5, for most of the multimodal test functions, the Mean value of the IAO is

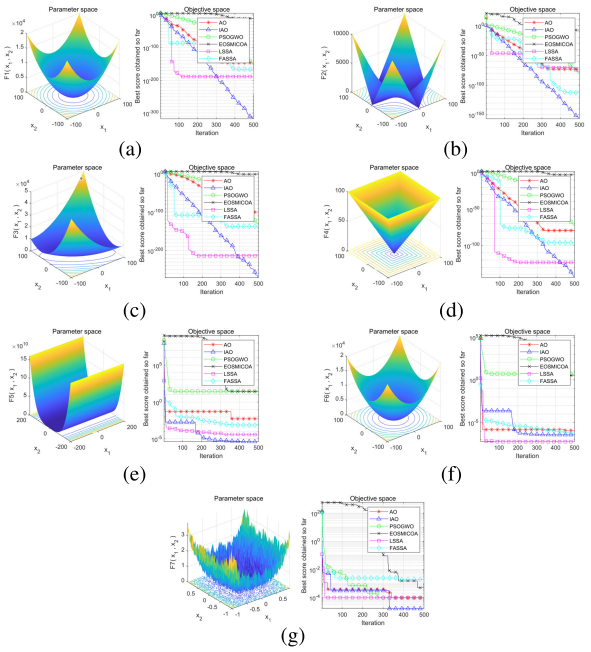

FIGURE 10. Convergence curves of different algorithms on F1-F7.

TABLE 4. Comparation results of different algorithms on single mode functions.

		AO	IAO	EOSMICOA	PSOGWO	LSSA	FASSA
F1	Best	5.67E-157	**0.00E+00**	4.82E-13	2.37E-159	0.00E+00	0.00E+00
	Worst	1.97E-120	**1.83E-284**	1.13E-10	8.83E-154	3.86E-145	1.47E-125
	Mean	1.32E-121	**1.24E-285**	3.42E-11	6.34E-155	2.57E-146	9.79E-127
	Std	5.09E-121	**0.00E+00**	3.82E-11	2.27E-154	9.96E-146	3.79E-126
F2	Best	3.38E-79	1.61E-169	6.12E-09	5.45E-83	0.00E+00	0.00E+00
	Worst	1.34E-51	**3.01E-149**	1.42E-07	3.40E-80	2.06E-43	2.06E-43
	Mean	8.93E-53	**2.03E-150**	5.32E-08	3.15E-81	1.37E-44	1.37E-44
	Std	3.46E-52	**7.77E-150**	4.07E-08	8.56E-81	5.32E-44	5.32E-44
F3	Best	2.31E-157	1.92E-312	6.59E-04	9.49E-138	0.00E+00	0.00E+00
	Worst	1.02E-99	**9.01E-260**	3.20E+01	8.77E-132	8.28E-109	1.99E-79
	Mean	7.36E-101	**6.00E-261**	3.55E+00	6.59E-133	5.52E-110	1.33E-80
	Std	2.61E-100	**0.00E+00**	8.24E+00	2.25E-132	2.14E-109	5.14E-80
F4	Best	3.17E-80	2.21E-160	1.55E-03	1.06E-75	0.00E+00	0.00E+00
	Worst	5.42E-53	**2.00E-135**	1.46E-01	8.18E-72	1.03E-68	1.11E-51
	Mean	3.61E-54	**1.33E-136**	3.02E-02	8.28E-73	6.89E-70	7.37E-53
	Std	1.40E-53	**5.17E-136**	4.41E-02	2.14E-72	2.67E-69	2.85E-52
F5	Best	2.06E-05	**3.22E-06**	2.70E+01	2.76E+01	2.85E-08	4.78E-06
	Worst	2.26E-02	2.80E-03	2.88E+01	2.87E+01	3.39E-04	2.33E-03
	Mean	4.22E-03	4.72E-04	2.78E+01	2.81E+01	9.70E-05	3.13E-04
	Std	6.66E-03	8.06E-04	6.12E-01	2.07E-01	1.08E-04	5.95E-04
F6	Best	1.24E-06	**7.65E-11**	2.05E+00	4.27E+00	9.49E-10	9.08E-10
	Worst	1.27E-03	**1.98E-06**	3.50E+00	4.99E+00	4.44E-06	2.45E-06
	Mean	1.91E-04	**3.15E-07**	2.79E+00	4.75E+00	6.99E-07	7.08E-07
	Std	3.52E-04	**4.96E-07**	4.56E-01	2.05E-01	1.49E-06	9.77E-07
F7	Best	7.87E-06	1.51E-05	4.96E-04	1.43E-05	7.93E-05	1.74E-04
	Worst	4.27E-04	**2.40E-04**	8.76E-03	6.44E-04	8.98E-04	2.92E-03
	Mean	9.43E-05	**1.30E-04**	2.90E-03	1.64E-04	4.19E-04	1.12E-03
	Std	1.10E-04	**7.45E-05**	2.19E-03	1.58E-04	2.78E-04	7.36E-04

significant lower than that of AO. On the F9, F10, F11 and F16 test functions, the Mean and Std. of the IAO can reach the optimum. Compared with other algorithms on the F13, F14 and F15 test functions, it can be found that the Mean and Std. of IAO are superior to other algorithms, which proves that IAO has excellent local search ability.

In order to evaluate the algorithm more comprehensively, the experiment also includes fixed-dimension multi-mode reference functions(F17-F23). The convergence curve of the comparison algorithms on the reference functions (F17-F23) is shown in Figure. 12.

It can be seen that the IAO algorithm converges quickly on most of the test functions, and the convergence position is very close to the optimal position of the test function.

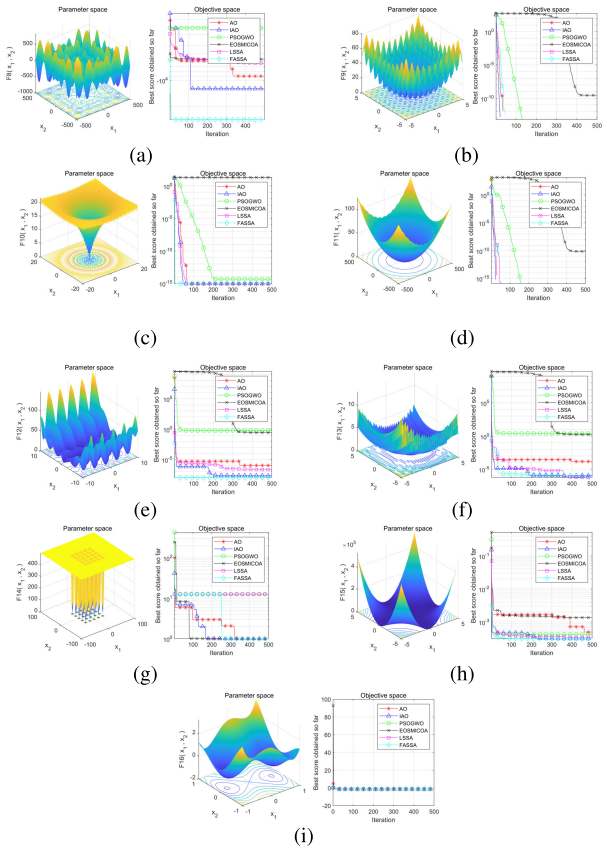

FIGURE 11. Convergence curves of different algorithms on F8-F16.

TABLE 5. Comparation results of different algorithms on multi-modal functions.

		AO	IAO	EOSMICOA	PSOGWO	LSSA	FASSA
F8	Best	-1.26E+04	**-1.26E+04**	-5.90E+03	-3.08E+03	-1.25E+04	-3.45E+04
	Worst	-3.80E+03	-4.72E+03	-5.64E+03	-2.02E+03	-5.55E+03	-2.35E+04
	Mean	-7.99E+03	**-9.04E+03**	-5.72E+03	-2.49E+03	-8.58E+03	-3.03E+04
	Std	4.03E+03	3.09E+03	7.05E+01	2.81E+02	2.44E+03	2.94E+03
F9	Best	0.00E+00	**0.00E+00**	2.69E-06	0.00E+00	0.00E+00	0.00E+00
	Worst	0.00E+00	**0.00E+00**	2.19E-07	0.00E+00	0.00E+00	0.00E+00
	Mean	0.00E+00	**0.00E+00**	6.95E-07	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	**0.00E+00**	6.95E-07	0.00E+00	0.00E+00	0.00E+00
F10	Best	8.88E-16	**8.88E-16**	2.00E+01	4.44E-15	8.88E-16	8.88E-16
	Worst	8.88E-16	**8.88E-16**	2.00E+01	4.44E-15	8.88E-16	8.88E-16
	Mean	8.88E-16	**8.88E-16**	2.00E+01	4.44E-15	8.88E-16	8.88E-16
	Std	0.00E+00	**0.00E+00**	9.86E-04	0.00E+00	0.00E+00	0.00E+00
F11	Best	0.00E+00	**0.00E+00**	2.95E-11	0.00E+00	0.00E+00	0.00E+00
	Worst	0.00E+00	**0.00E+00**	5.01E-02	0.00E+00	0.00E+00	0.00E+00
	Mean	0.00E+00	**0.00E+00**	3.42E-03	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	**0.00E+00**	1.29E-02	0.00E+00	0.00E+00	0.00E+00
F12	Best	4.65E-08	1.35E-11	1.69E-01	5.10E-01	6.60E-10	3.26E-11
	Worst	3.99E-05	1.93E-06	3.87E-01	6.45E-01	5.97E-07	8.11E-07
	Mean	4.65E-06	3.53E-07	2.44E-01	5.69E-01	1.02E-07	8.70E-08
	Std	1.01E-05	**5.91E-07**	5.09E-02	4.44E-02	1.58E-07	2.20E-07
F13	Best	7.78E-07	2.41E-11	1.32E+00	1.98E+00	1.83E-10	1.05E-12
	Worst	9.28E-05	1.87E-06	1.95E+00	2.64E+00	1.08E-05	4.56E-06
	Mean	2.56E-05	2.15E-07	1.67E+00	2.46E+00	1.16E-06	4.41E-07
	Std	2.98E-05	**4.74E-07**	1.90E-01	1.64E-01	2.93E-06	1.18E-06
F14	Best	9.98E-01	**9.98E-01**	9.98E-01	9.99E-01	9.98E-01	9.98E-01
	Worst	2.98E+00	**1.27E+01**	2.98E+00	1.27E+01	1.27E+01	1.27E+01
	Mean	1.66E+00	2.82E+00	1.13E+00	4.80E+00	8.60E+00	7.43E+00
	Std	8.10E-01	4.04E+00	5.12E-01	4.40E+00	4.96E+00	5.59E+00
F15	Best	3.71E-04	**3.07E-04**	1.23E-03	3.55E-04	3.08E-04	3.07E-04
	Worst	6.15E-04	5.86E-04	1.35E-03	6.08E-04	3.36E-04	1.22E-03
	Mean	4.86E-04	3.44E-04	1.27E-03	4.54E-04	3.14E-04	3.71E-04
	Std	7.50E-05	8.41E-05	3.38E-05	7.79E-05	8.91E-06	2.36E-04
F16	Best	-1.03E+00	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Worst	-1.03E+00	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Mean	-1.03E+00	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	3.09E-04	**3.14E-16**	1.21E-05	1.03E-04	7.94E-11	3.31E-10

According to Table 6, the convergence Mean and Std. of IAO algorithm on test functions F17, F18, F22 and F23 have reached the optimal position among all algorithms.

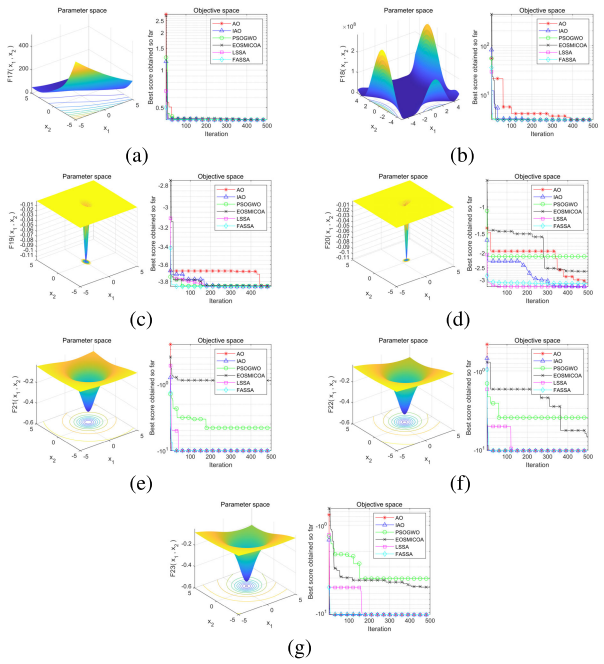

FIGURE 12. Convergence curves of different algorithms on F17-F23.

TABLE 6. Comparation results of different algorithms on fixed-dimension multi-mode reference function.

		AO	IAO	EOSMICOA	PSOGWO	LSSA	FASSA
F17	Best	3.98E-01	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Worst	3.98E-01	**3.98E-01**	5.04E+00	4.08E-01	3.98E-01	3.98E-01
	Mean	3.98E-01	**3.98E-01**	1.02E+00	4.02E-01	3.98E-01	3.98E-01
	Std	2.53E-04	**4.62E-12**	1.63E+00	3.00E-03	2.77E-09	4.75E-10
F18	Best	3.00E+00	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Worst	3.12E+00	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Mean	3.03E+00	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Std	3.04E-02	**1.03E-14**	4.64E-05	1.07E-04	3.06E-09	9.70E-09
F19	Best	-3.86E+00	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	Worst	-3.86E+00	**-3.86E+00**	-3.85E+00	-3.76E+00	-3.86E+00	-3.86E+00
	Mean	-3.86E+00	**-3.86E+00**	-3.85E+00	-3.76E+00	-3.86E+00	-3.86E+00
	Std	7.65E-03	2.62E-04	4.79E-04	2.55E-02	2.00E-01	2.84E-06
F20	Best	-3.27E+00	**-3.32E+00**	-3.07E+00	-2.88E+00	-3.32E+00	-3.32E+00
	Worst	-2.94E+00	-2.99E+00	-1.46E+00	-2.00E+00	-3.16E+00	-3.14E+00
	Mean	-3.12E+00	**-3.25E+00**	-2.24E+00	-2.51E+00	-3.28E+00	-3.25E+00
	Std	9.76E-02	**9.96E-02**	6.13E-01	2.75E-01	6.83E-02	7.13E-02
F21	Best	-1.02E+01	**-1.02E+01**	-5.03E+00	-6.09E+00	-1.02E+01	-1.02E+01
	Worst	-1.01E+01	**-1.02E+01**	-8.81E-01	-3.13E+00	-1.02E+01	-1.02E+01
	Mean	-1.01E+01	**-1.02E+01**	-2.53E+00	-4.14E+00	-1.02E+01	-1.02E+01
	Std	1.90E-02	1.62E-05	2.08E+00	7.46E-01	3.68E-06	3.32E-06
F22	Best	-1.04E+01	**-1.04E+01**	-6.88E+00	-5.48E+00	-1.04E+01	-1.04E+01
	Worst	-1.04E+01	**-1.04E+01**	-5.21E-01	-3.42E+00	-1.04E+01	-1.04E+01
	Mean	-1.04E+01	**-1.04E+01**	-4.04E+00	-4.05E+00	-1.04E+01	-1.04E+01
	Std	9.27E-03	2.07E+00	2.07E+00	5.48E-01	2.28E-06	7.21E-06
F23	Best	-1.05E+01	**-1.05E+01**	-5.13E+00	-4.77E+00	-1.05E+01	-1.05E+01
	Worst	-1.04E+01	**-1.05E+01**	-9.44E-01	-3.53E+00	-1.04E+01	-1.04E+01
	Mean	-1.05E+01	**-1.05E+01**	-3.44E+00	-4.03E+00	-1.05E+01	-1.05E+01
	Std	2.78E-02	2.34E-05	2.10E+00	3.97E-01	5.64E-06	8.05E-07

Based on the above analysis, it can be seen that IAO has good convergence and fast convergence speed. Meanwhile, the convergence mean value and the convergence average value maintain a small level on most of the test functions, indicating that the stability of IAO is good.

B. SIMULATION EXPERIMENTS FOR PATH PLANNING

This paper designs a path planning method for the obstacle avoidance by using an optimized interval type-2 fuzzy controller. In order to fully verify the path planning effect of the IAOFC algorithm and to ensure the reliability and robustness of it. Simulation experiments in static environment and dynamic environment were respectively conducted. The static simulation environment is shown in Figure. 13.

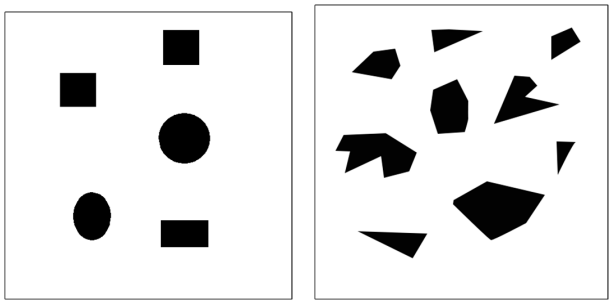

FIGURE 13. Static simulation environments.

Setting the dynamic simulation environment as shown in Figure. 14, the obstacle is represented by the shape of a ''car''. In the simulation environment, the obstacle moves uniformly in one direction.

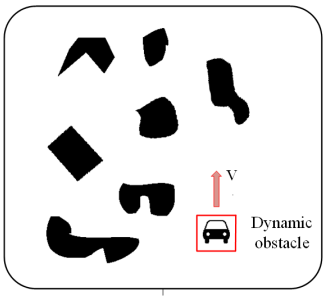

FIGURE 14. Dynamic simulation environment.

In the interval type-2 fuzzy controller, IAO is used to optimize the membership function of the system, and the cost function is constructed according to the path planning length and time, which are normalized. The cost function is defined by

$$fitness = t + s \tag{31}$$

where t represents the time spent in the path planning and s represents the length of the path planning. The optimization is carried out on different maps, and the number of populations is set to 50, the number of iterations is set to 100. The parameters of the optimized membership function are shown in Table 7.

The optimized membership function image is shown in Figure. 15.

The simulation results in static environment are shown in Figure. 16.

In Figure. 16(a) is the path planning of the robot in a simple static environment, Figure. 16(b) is the path planning of the robot in a complex static environment, Figure. 16(c) is the iterative value of each fuzzy variable when the fuzzy controller performs path planning, and Figure. 16(d) is the iterative value of each fuzzy variable when the fuzzy controller performs path planning in a complex environment. As can be seen from Figure. 15(c), the variation of the

TABLE 7. Optimized membership function parameters.

Fuzzy Variable	Fuzzy Subset	Membership Function	Function Parameters
DistanceFront	Low	Gaussmf	[0.081 0.123],[0.9175],[0.449 0.4492]
	Medium	Gaussmf	[0.4004 0.2991],[0.963],[0.2539 0.2539]
	Large	Sigmf	[20.294 0.452],[0.804],[0.9295 0.9295]
DistanceFrontLeft	Small	Gaussmf	[0.0323 0.138723],[0.7624],[0.93 0.93]
	Medium	Gaussmf	[0.0268 0.1532],[0.6830],[0.795 0.795]
DistanceFrontRight	Small	Gaussmf	[0.0907 0.030],[0.6084],[0.6602 0.6602]
	Medium	Gaussmf	[0.0335 0.1958],[0.779],[0.97 0.97]
Anglegoal	Negative	Gaussmf	[0.0103 -0.34],[0.448],[0.36 0.36]
	No	Gaussmf	[0.06177 0.1529],[0.57],[0.575 0.575]
	Positive	Gaussmf	[0.6675 0.5721],[0.4081],[0.28 0.28]
	MoreNegative	Sigmf	[-23.24 -0.413],[0.943],[0.32 0.32]
	MorePositive	Sigmf	[30 0.8],[0.812],[0.8608 0.8608]
Turn	Left	Trimf	[-2.5 -1.186 -0.6],[0.39],[0.28 0.28]
	Right	Trimf	[0.381 1.158 1.846],[0.872],[0.454 0.454]
DistanceGoal	Low	Gaussmf	[0.0568 0.1624],[0.738],[0.89 0.89]
	Medium	Gaussmf	[0.0504 0.2441],[0.951],[0.303 0.303]
	High	Sigmf	[22.01 0.533],[0.967],[0.237 0.237]
Steer	Left	Gaussmf	[0.03 -0.26],[0.74],[0.87 0.87]
	No	Gaussmf	[0.039 0.027],[0.832],[0.6272 0.6272]
	Right	Gaussmf	[0.0379 0.262],[0.7972],[0.738 0.738]
	MoreLeft	Gaussmf	[0.566 -0.476],[0.67],[0.87 0.87]
	MoreRight	Gaussmf	[0.04312 0.6464],[0.916],[0.439 0.439]

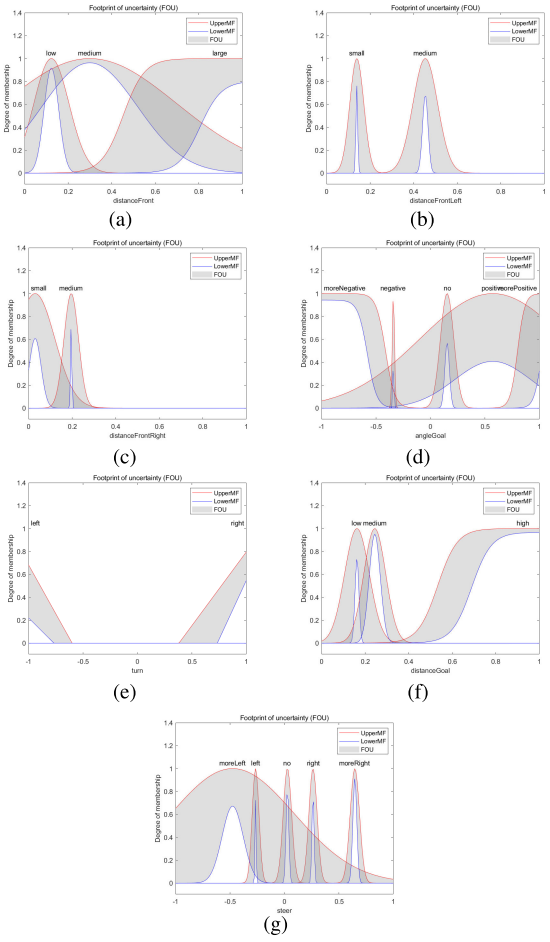

FIGURE 15. Optimized membership function curve.

variables of the fuzzy controller is smoother, compared to Figure. 15(d). In the simple static environment, the shape of obstacles is more regular, and the difficulty of path planning is relatively small. As can be seen from Figure. 15(d), the variable range of obstacle distance, steering and other variables is larger, and the curve fluctuation is larger, which indicates that the fuzzy controller can give a more positive

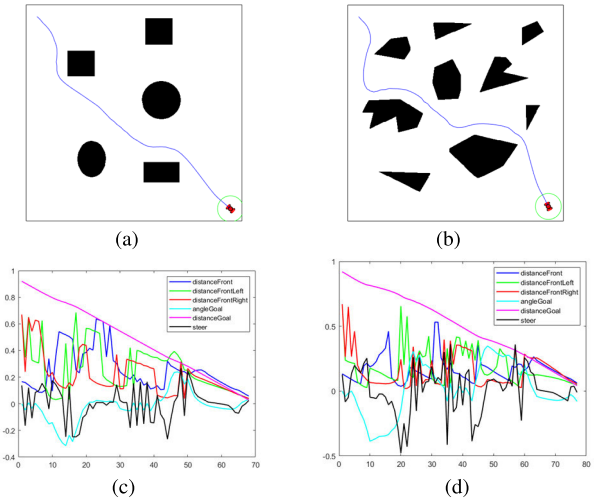

FIGURE 16. Path planning results in static environment.

response in the face of complex environment, and the robustness and adaptability of the algorithm are better under complex environment.

In the face of more uncertainties, the experiment was designed for the robot to perform path planning in a dynamic environment. The design of the dynamic environment is shown in Figure. 16(a). The dynamic obstacle is represented by a "car" in the red box, and the obstacle moves forward at the speed of $v = 2$ on the robot's path to the target.

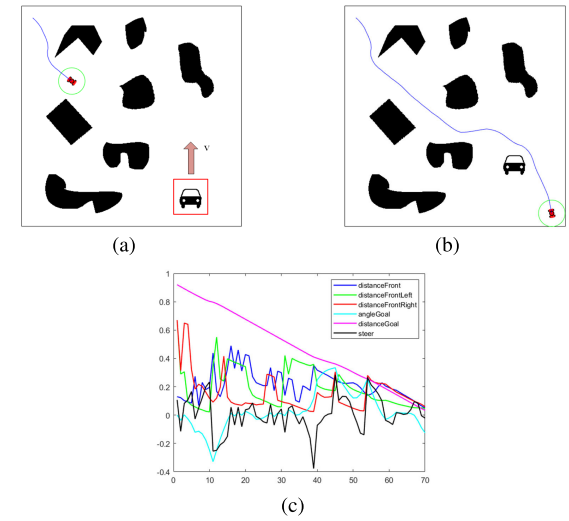

FIGURE 17. Path planning results in dynamic environment.

In Figure. 17(a) and Figure. 17(b) represent the path planning of the robot in the dynamic environment, and Figure. 17(c) represent the iterative changes of various variables during the path planning of the robot. In the dynamic environment, there are not only irregular obstacles, but also moving obstacles that interfere with the path planning. It can be seen from Figure. 17(c) that the fuzzy variable curve has a

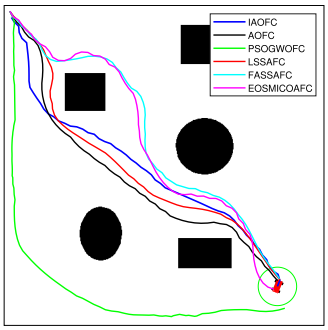

FIGURE 18. Comparison of various path planning algorithms in static environment.

TABLE 8. Comparison results of different path planning algorithms.

Path planning algorithm	Start	Goal	Length Cost	Time Cost
IAOFC	[10,10]	[450,450]	851.67	2.81
AOFC	[10,10]	[450,450]	846.54	4.41
PSOGWOFC	[10,10]	[450,450]	855.75	5.95
LSSAFC	[10,10]	[450,450]	854.67	3.35
FASSAFC	[10,10]	[450,450]	875.32	3.15
EOSMICOAFC	[10,10]	[450,450]	897.88	5.12

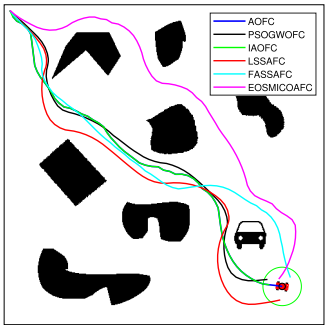

FIGURE 19. Comparison of various path planning algorithms in dynamic Environment.

TABLE 9. Comparison results of different path planning algorithms.

Path planning algorithm	Start	Goal	Length Cost	Time Cost
IAOFC	[10,10]	[450,450]	854.67	3.35
AOFC	[10,10]	[450,450]	866.81	3.39
PSOGWOFC	[10,10]	[450,450]	855.12	4.17
LSSAFC	[10,10]	[450,450]	838.84	3.6
FASSAFC	[10,10]	[450,450]	850.69	3.75
EOSMICOAFC	[10,10]	[450,450]	879.97	4.02

large change and its slope has a large change, which indicates that the fuzzy controller responds quickly to the dynamic environment, reflecting the adaptability of fuzzy control to nonlinear problems.

To further verify the performance of the IAOFC algorithm, we use the fuzzy controller optimized by other swarm intelligence algorithms. The experiment is firstly conducted in a static environment. Each algorithm is repeatedly run by ten times, and the final results are averaged. The experimental results are shown in Figure. 18, and the specific path planning length cost and time cost are shown in Table 8.

As shown in Figure. 18, the path planning of most algorithms is in the middle of the map, while the path of PSOGWOFC algorithm deviates from the shortest path and falls into the local optimization. Compared with other algorithms, the path planning of the IAOFC algorithm maintains a safer distance from obstacles, and the path is relatively smooth. Combined with Table 8, it can be seen that IAOFC has the lowest path planning cost and time cost in the experiment, which proves the effectiveness of IAO algorithm in fuzzy controller optimization.

The result of path planning in dynamic environment is shown in Figure 18, and the specific path cost and planning time cost are shown in Table 9.

According to the results of Table 9, although the planning time of IAOFC algorithm in dynamic environment is not the shortest, the path length is the best. Synthetically considering them, the path planning performance of IAOFC algorithm is still the best among the comparison algorithms.

VI. CONCLUSION

Aiming at the problem of slow computation speed and low efficiency of traditional path planning algorithm, due to the complexity and uncertainty of environment in the local path planning, this paper proposes a path planning method based on interval type-2 fuzzy control. The path planning method transforms the environment information into physical information and maps it to the fuzzy domain, and the physical information of robot motion can be obtained by fuzzy solution. This method does not require accurate mathematical modeling, but has fast response speed and high path planning efficiency. Second, an improved Aquila Optimizer (IAO) is proposed, which combines the AO with the spiral foraging strategy of WOA and the hazard warning mechanism of SSA, and uses the opposition-based learning strategy fused with Cauchy mutation to perturb the cruise results, so that the algorithm can jump out of the local region more easily. The IAO is applied to 23 classical test functions to prove that it is better for the global optimization. Finally, the IAO is used to optimize the membership function parameters of an interval type-2 fuzzy controller, and the simulation and comparison experiments with other algorithms are carried out in static and dynamic environments. According to comparison results of different path planning algorithms, if the path cost and time cost are normalized and added together as the total cost of the algorithm, the proposed IAOFC algorithm reduces the cost by an average of about 15% compared to other algorithms in a static environment, and an average of about 6% in a dynamic environment.

In future research, further research will be conducted on multi-objective optimization performance of fuzzy controllers to achieve the overall optimal performance of the control algorithm.

APPENDIX

CODE OF IAOFC

Partial code of the optimized controller design is given in Appendix.

```
%initialize
X0=initializationNew(N,Dim,UB,LB);
X=X0;
%Calculate initial fitness value
Ffun=zeros(1,size(X,1));
for i = 1:size(X,1)
    fuz=writemf(X(i,:),fuz);
    writeFIS(fuz,'IAOfuzzy');
    Ffun(i)=FitnessCalculate(X(i,:));
end
%Sort
[Ffun, index]= sort(Ffun);
WorstF = Ffun(end);Best_FF = Ffun(1);
for i = 1:size(X,1)
   X(i,:) = X0(index(i),:);
end
Best_P = X(1,:);
Xnew = X;t=1;
Ffun_new=zeros(1,size(Xnew,1));

while t<T+1
     for i=1:size(X,1)
        %boundary control
        X(i,:)=BC(X(i,:));
        %Update Location
        fuz=writemf(X(i,:),fuz);
        writeFIS(fuz,'IAOfuzzy');
        Ffun(1,i)=
        FitnessCalculate(X(i,:));
        [Best_FF,Best_P]=
        GreedyAl(Ffun(1,i),Best_FF,Best_P);
     end
for i=1:size(X,1)
    if t<=(2/3)*T
        if rand <0.5
            Xnew(i,:)=eq_15(X(i,:),
            Best_P(1,:),t,T);
            Xnew(i,:)=BC(X(i,:));
            %Update Location
            fuz=writemf(Xnew(i,:),fuz);
            writeFIS(fuz,'IAOfuzzy');
            Ffun_new(1,i)=
            FitnessCalculate(Xnew(i,:));
            [Ffun(1,i),X(i,:)]=
            GreedyAl(Ffun(1,i),
            Ffun_new(1,i),Xnew(i,:));
        else
            Xnew(i,:)=eq_18(X(i,:),
            Best_P(1,:));
            Xnew(i,:)=BC(X(i,:));
            %Update Location
            fuz=writemf(Xnew(i,:),fuz);
            writeFIS(fuz,'IAOfuzzy');
            Ffun_new(1,i)=
            FitnessCalculate(Xnew(i,:));
            [Ffun(1,i),X(i,:)]=
            GreedyAl(Ffun(1,i),
            Ffun_new(1,i),Xnew(i,:));
        end
else
    if rand<0.5
      Xnew(i,:)=eq_25(X(i,:),
        Best_P(1,:));
      %boundary control
      Xnew(i,:)=BC(X(i,:));
      %Update Location
      fuz=writemf(Xnew(i,:),fuz);
      writeFIS(fuz,'IAOfuzzy');
      Ffun_new(1,i)=
      FitnessCalculate(Xnew(i,:));
      [Ffun(1,i),X(i,:)]=
        GreedyAl(Ffun(1,i),
        Ffun_new(1,i),Xnew(i,:));
    else
     Xnew(i,:)=eq_25(X(i,:),
        Best_P(1,:));
     end
     %boundary control
     Xnew(i,:)=BC(X(i,:));
     %Update Location
     fuz=writemf(Xnew(i,:),fuz);
     writeFIS(fuz,'newIAOfuzzy');
     Ffun_new(1,i)=
     FitnessCalculate(Xnew(i,:));
     [Ffun(1,i),X(i,:)]=
        GreedyAl(Ffun(1,i),
        Ffun_new(1,i),Xnew(i,:));
   end
    for j = 1:size(X,1)
       if(Ffun(j) < Best_FF)
         Best_FF = Ffun(j);
         Best_P = X(j,:);
       end
    end
    %sort
    [Ffun, index]= sort(Ffun);
    WorstF = Ffun(end);
    Best_FF = Ffun(1);
    for j = 1:size(X,1)
       X(j,:) = X(index(j),:);
    end
end
%sort
[Ffun, index]= sort(Ffun);
BestF = Ffun(1);
WorstF = Ffun(end);
for j = 1:size(X,1)
   X(j,:) = X(index(j),:);
end
conv(t)=Best_FF;
t=t+1;
fuz=writemf(Temp,fuz);
writeFIS(fuz,'IAOfuzzy');
Ffun_Temp=FitnessCalculate(Temp);
```

```
    if(Ffun_Temp<Best_FF)
        Best_P =Temp;
        Best_FF = Ffun_Temp;
    end
    [Best_FF,Best_P]=RC(Temp,Ffun_Temp);
    [Best_FF,Best_P]=GreedyAl(Ffun(1,i),
    Ffun_Temp,Temp);
end
fuz=writemf(Best_P,fuz);
writeFIS(fuz,'IAOfuzzy');
```

ACKNOWLEDGMENT

The authors would like to thank the Faculty of Electrical and Control Engineering, Liaoning Technical University.

REFERENCES

[1] D. Wang, S. Chen, Y. Zhang, and L. Liu, "Path planning of mobile robot in dynamic environment: Fuzzy artificial potential field and extensible neural network," *Artif. Life Robot.*, vol. 26, no. 1, pp. 129–139, Feb. 2021.

[2] M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1102–1110, Jun. 2009.

[3] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[4] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[5] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[7] Q. Liu, Y. Zhang, M. Li, Z. Zhang, N. Cao, and J. Shang, "Multi-UAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network," *IEEE Access*, vol. 9, pp. 124670–124681, 2021.

[8] Y. Liu, Z. Zheng, F. Qin, X. Zhang, and H. Yao, "A residual convolutional neural network based approach for real-time path planning," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108400.

[9] X.-H. Liu, D. Zhang, J. Zhang, T. Zhang, and H. Zhu, "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm," *Cluster Comput.*, vol. 24, pp. 1–15, Jan. 2021.

[10] G. Wang and J. Zhou, "Dynamic robot path planning system using neural network," *J. Intell. Fuzzy Syst.*, vol. 40, no. 2, pp. 3055–3063, Feb. 2021.

[11] X. Cao, L. Chen, L. Guo, and W. Han, "AUV global security path planning based on a potential field bio-inspired neural network in underwater environment," *Intell. Autom. Soft Comput.*, vol. 27, no. 2, pp. 391–407, 2021.

[12] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.

[14] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, Jul. 1997.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[16] R. Lotfi, B. Kargar, M. Rajabzadeh, F. Hesabi, and E. Özceylan, "Hybrid fuzzy and data-driven robust optimization for resilience and sustainable health care supply chain with vendor-managed inventory approach," *Int. J. Fuzzy Syst.*, vol. 24, no. 2, pp. 1216–1231, Mar. 2022.

[17] N. Abdenouri, A. Zoukit, I. Salhi, and S. Doubabi, "Model identification and fuzzy control of the temperature inside an active hybrid solar indirect dryer," *Sol. Energy*, vol. 231, pp. 328–342, Jan. 2022.

[18] X. Bu, Q. Qi, and B. Jiang, "A simplified finite-time fuzzy neural controller with prescribed performance applied to waverider aircraft," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2529–2537, Jul. 2022.

[19] L. Zhang, B. Chen, C. Lin, and Y. Shang, "Fuzzy adaptive fixed-time consensus tracking control of high-order multiagent systems," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 2, pp. 567–578, Feb. 2022.

[20] P. A. Ejegwa, S. Wen, Y. Feng, W. Zhang, and N. Tang, "Novel Pythagorean fuzzy correlation measures via Pythagorean fuzzy deviation, variance, and covariance with applications to pattern recognition and career placement," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 1660–1668, Jun. 2022.

[21] L. Hao, J. H. Park, and D. Ye, "Fuzzy logic systems-based integral sliding mode fault-tolerant control for a class of uncertain non-linear systems," *IET Control Theory Appl.*, vol. 10, no. 3, pp. 300–311, Feb. 2016.

[22] T.-L. Lee and C.-J. Wu, "Fuzzy motion planning of mobile robots in unknown environments," *J. Intell. Robot. Syst.*, vol. 37, no. 2, pp. 177–191, Jun. 2003.

[23] Q. Song, Q. Zhao, S. Wang, Q. Liu, and X. Chen, "Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization," *IEEE Access*, vol. 8, pp. 62107–62115, 2020.

[24] V. N. Naik and S. P. Singh, "A novel interval type-2 fuzzy-based direct torque control of induction motor drive using five-level diode-clamped inverter," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 149–159, Jan. 2021.

[25] M. H. Khooban and M. Gheisarnejad, "A novel deep reinforcement learning controller based type-II fuzzy system: Frequency regulation in microgrids," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 689–699, Aug. 2021.

[26] E. Sohrabzadi, M. Gheisarnejad, Z. Esfahani, and M. H. Khooban, "A novel intelligent ultra-local model control-based type-II fuzzy for frequency regulation of multi-microgrids," *Trans. Inst. Meas. Control*, vol. 44, no. 5, pp. 1134–1148, Mar. 2022.

[27] H. Acikgoz, R. Coteli, E. Tanyildizi, B. Dandil, and K. Kayisli, "Advanced control of three-phase PWM rectifier using interval type-2 fuzzy neural network optimized by modified golden sine algorithm," *Electr. Power Compon. Syst.*, vol. 51, no. 10, pp. 933–948, Jun. 2023.

[28] P. Maji and P. Garai, "Rough hypercuboid based generalized and robust IT2 fuzzy C-Means algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3641–3652, Jul. 2021.

[29] M. Dirik, O. Castillo, and A. F. Kocamaz, "Visual-servoing based global path planning using interval type-2 fuzzy logic control," *Axioms*, vol. 8, no. 2, p. 58, May 2019.

[30] L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: A novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, vol. 157, Jul. 2021, Art. no. 107250.

[31] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Oct. 2000.

[32] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[33] N. F. Jamin, N. M. A. Ghani, Z. Ibrahim, A. N. K. Nasir, M. Rashid, and M. O. Tokhi, "Stabilizing control of two-wheeled wheelchair with movable payload using optimized interval type-2 fuzzy logic," *J. Low Freq. Noise, Vib. Act. Control*, vol. 40, no. 3, pp. 1585–1606, Sep. 2021.

[34] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commerce (CIMCA-IAWTIC)*, vol. 1, 2005, pp. 695–701.

[35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[36] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.

[37] H. M. Rabie, "Particle swarm optimization and grey wolf optimizer to solve continuous p-median location problems," in *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges* (Studies in Big Data), vol. 77. Cham, Switzerland: Springer, Dec. 2020, pp. 415–435.

[38] D. Chen, J. Zhao, P. Huang, X. Deng, and T. Lu, "An improved sparrow search algorithm based on levy flight and opposition-based learning," *Assem. Autom.*, vol. 41, no. 6, pp. 697–713, Nov. 2021.

[39] Q. Huang, S. Liu, M. M. Li, and Y. X. Guo, "Multi-strategy chimp optimization algorithm and its application of engineering problem," *Comput. Eng. Appl.*, vol. 58, no. 19, pp. 174–183, 2022.
[40] L. Shi, X. Ding, M. Li, and Y. Liu, "Research on the capability maturity evaluation of intelligent manufacturing based on firefly algorithm, sparrow search algorithm, and BP neural network," *Complexity*, vol. 2021, pp. 1–26, Aug. 2021.
[41] Q. He, J. Lin, and H. Xu, "Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm," *Control Decis.*, vol. 36, no. 7, p. 11, 2021.

KUN LI received the B.Sc. degree from the Shandong University of Science and Technology, in 2005, the M.Sc. degree from Liaoning Technical University, in 2008, and the Ph.D. degree from Northeastern University, in 2013. He is currently a Professor with the Department of Electrical Engineering and Intelligent Control, Faculty of Electrical and Control Engineering, Liaoning Technical University. His current research interests include industrial artificial intelligence methods and applications.

XIANG ZHANG received the B.Sc. degree from Jinzhong University, in 2019. He is currently pursuing the M.Sc. degree with Liaoning Technical University, Huludao, China. His current research interests include intelligent optimal control and their applications.

YING HAN received the B.Sc. and M.Sc. degrees from Liaoning Technical University, in 2005 and 2008, respectively, and the Ph.D. degree from Northeastern University, China, in 2020. She is currently an Associate Professor with the Faculty of Electrical and Control Engineering, Liaoning Technical University. Her research interests include time series analysis and applications, and complex system prediction.