symmetry

MDPI

Article

Automatic Parking Path Optimization Based on Immune Moth Flame Algorithm for Intelligent Vehicles

Yan Chen [1], Longda Wang [2,*], Gang Liu [3,4] and Bing Xia [5]

[1] School of Mechanical and Electrical Engineering, Chizhou University, Chizhou 247000, China
[2] School of Automation and Electrical Engineering, Dalian Jiaotong University, Dalian 116026, China
[3] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
[4] School of Mechanical Engineering, Inner Mongolia University for the Nationalities, Tongliao 028000, China
[5] Hangzhou Huayun Technology Co., Ltd., Hangzhou 310012, China
* Correspondence: ldwangdl@sina.com; Tel.: +86-132-3690-1631

Abstract: Automatic parking path optimization is a key point for automatic parking. However, it is difficult to obtain the smooth, accurate and optimal parking path by using traditional automatic parking optimization algorithms. So, based on the automatic parking path optimization model for cubic spline interpolation, an improved automatic parking path optimization based on the immune moth flame algorithm is proposed for intelligent vehicles. Firstly, to enhance the global optimization performance, an automatic parking path optimization model for cubic spline interpolation is designed by using shortest parking path as optimization target. Secondly, an improved immune moth flame algorithm (IIMFO) based on the immune mechanism, Gaussian mutation mechanism and opposition-based learning strategy is proposed, and an adaptive decreasing inertia weight coefficient is integrated into the moth flame algorithm so that these strategies can improve the balance quality between global search and local development effectively. Finally, the optimization results on the several common test functions show that the IIMFO algorithm proposed in this paper has higher optimization precision. Furthermore, the simulation and semi-automatic experiment results of automatic parking path optimization practical cases show that the improved automatic parking path optimization based on the immune moth flame algorithm for intelligent vehicles has a better optimization effect than that of the traditional automatic parking optimization algorithms.

Keywords: cubic spline interpolation; automatic parking; path optimization; moth flame algorithm; immune mechanism

check for updates

Citation: Chen, Y.; Wang, L.; Liu, G.; Xia, B. Automatic Parking Path Optimization Based on Immune Moth Flame Algorithm for Intelligent Vehicles. *Symmetry* **2022**, *14*, 1923. https://doi.org/10.3390/sym14091923

Academic Editor: Calin Ciufudean

Received: 22 July 2022
Accepted: 7 September 2022
Published: 14 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The automatic parking path optimization algorithm for intelligent vehicles is a significant means for ensuring safe, comfortable, rapid and accurate parking. The ideal intelligent automatic parking path optimization algorithm should meet the following requirements. Once the parking feasibility conditions are met, a smooth automatic parking path without collision avoidance, with a length as short as possible, will be given efficiently.

Considering the performance in automatic parking, various relative researchers' studies have been proposed in the recent literature. Among them, a reinforcement learning-based end-to-end parking algorithm was proposed to realize automatic parking [1]. An exponential (epsilon-convergent) control algorithm for chained systems was proposed, and it was verified to the automatic parking system in [2]. A new adaptive robust four-wheel positioning method was proposed, and it was verified to the automatic parking system in [3]. A model-based reinforcement learning method was proposed to learn parking strategies from the data [4]. A structure of deep neural network-based control for automatic parking maneuver process was designed and implemented [5]. Obviously, the existing studies can improve the control effect for automatic parking control.

The automatic parking path optimization problem is characterized by an unclear internal mechanism, numerous uncertain factors, etc. Because traditional optimization algorithms can easily result in a local optimum, obtaining satisfactory methods is difficult. To solve the aforementioned issue in automatic traditional optimization algorithms, much of the literature has discussed the improved strategies with traditional optimization algorithms. A hybrid genetic algorithm was proposed, which is composed of two algorithms: genetic and classic in [6]. An adaptive differential evolution algorithm was proposed in [7], which selects new mutation and crossover strategies for global search optimization. An evolutionary multi-objective seagull optimization algorithm for global optimization (EMOSOA) was proposed in [8]. A modification of the nature-inspired symbiotic search (MSOS) algorithm was proposed to improve the accuracy of its search and exploration by introducing adaptive benefit factors and modified parasitic vectors in [9]. A heat transfer search (HTS) algorithm was proposed to elucidate structural optimization problems with multi-objective functions (called MOHTS) in [10]. In addition, the improved ant colony optimization (ACO) for path planning in an automated guided vehicle (AGV)-based intelligent parking system was proposed in [11].

The moth flame algorithm (MFO) is one of the most efficient intelligent algorithms, and many works in the literature have proposed various improved moth flame algorithms. In order to determine the optimal multilevel thresholding for image segmentation, the quality of the whale optimization algorithm (WOA) and moth flame optimization (MFO) were verified in [12]. A optimal kernel extreme learning machine (KELM) using a chaotic moth flame optimization (CMFO) was proposed in [13]. A hybrid learning machine using a new moth flame optimization was proposed in [14]. A new moth flame optimization algorithm using confrontation learning and position updating mechanism was proposed in [15]. A moth flame optimization algorithm (IMFO) was proposed to be applied to the engineering practice in [16]. A binary moth flame optimization (B-MFO) algorithm for feature selection from medical datasets was proposed in [17]. A moth flame optimization algorithm based on multi-trial vector (MV-MFO) was proposed in [18]. An improved version of the MFO algorithm based on Lévy flight strategy (LMFO) was proposed in [19]. An enhanced moth flame optimization (EMFO) technique based on cultural learning (CL) and Gaussian mutation (GM) was proposed in [20]. However, there is a little related literature published about the automatic parking path optimization method based on the improved immune moth flame algorithm.

Compared with studies existing about automatic parking path optimization method, the main innovations of this paper are as follows:

(I) Innovations of automatic parking path optimization method: Aiming at the problem that the optimal automatic parking path selected by the optimization method is not ideal for path length and the tracking trajectory is not smooth enough, a novel automatic parking path optimization model by using cubic spline interpolation is constructed. In this way, the complex automatic parking trajectory optimization problem can be simplified.

(II) Innovations of MFO: To solve the problem that traditional intelligent algorithms are prone to local convergence, this paper proposes an IIMFO. This method solves the defect that MFO lacks the mechanism of jumping out of local convergence, thus it effectively improves the balance quality between global search and local development, and it includes novel improvement strategies proposed. The specific novel strategies are described as follows: I introduce an immune mechanism in the iterative process and expand the search space; II introduce Gaussian mutation mechanism to further improve the ability of local development and anti-local convergence; III introduce adaptive decreasing weight coefficient, so as to further strengthen local search quality; and IV introduce OBL mechanism for elite to further improve optimization quality. The numerical results of the test functions indicate the improved strategies can improve to a certain area for optimization speed and accuracy. Additionally, the simulation and semi-automatic experiment results of the automatic parking path optimization practical cases indicate the above facts. The specific diagrammatic sketch about the improved strategies is shown in Figure 1.

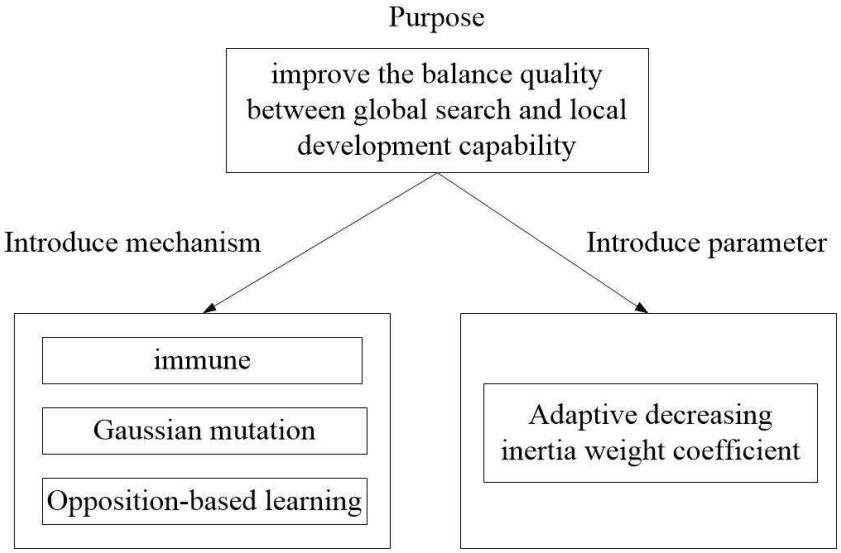


Figure 1. Diagrammatic sketch about improved strategies.

This paper is organized as follows. Section 2 introduces the automatic parking path optimization method using cubic spline interpolation. Section 3 introduces the MFO. Section 4 introduces the IIMFO. Section 5 uses test functions, simulation and semi-automatic experiment comparison based on intelligent automatic parking, and carries out an analysis. Section 6 concludes this article.

2. Automatic Parking Path Optimization Method Using Cubic Spline Interpolation

2.1. Cubic Spline Interpolation

The spline is originally a drawing tool used in engineering design. It is a kind of elastic thin wooden strip, which is used to describe the smooth shape curve in the manufacturing process of aircraft or ships. When used, it is fixed in several close points with a pressing iron and left to bend naturally in other places then slightly adjusted to make the curve conform to a satisfactory shape. Thus a spline curve is formed, and the mathematical model abstracted from it is called spline interpolation [21].

The specific formula of cubic spline interpolation is described below:

The interval $\Delta = [a, b]$ and its $n + 1$ interpolation points $(x_0, x_1, \cdots\cdots, x_n)$ are known, if the following two conditions of interpolation function $s(x)$ can be satisfied. The function $s(x)$ is a cubic spline interpolation function for interval $\Delta = [a, b]$.

Condition 1: in any arbitrary subinterval $[x_i, x_{i+1}]$, the index $i \in [0, 1, 2, \ldots, n]$, $s(x)$ is a cubic polynomial.

$$s(x) = \begin{cases} s_1(x) = d_1x^3 + c_1x^2 + b_1x + a_1 \in [x_0, x_1] \\ s_2(x) = d_2x^3 + c_2x^2 + b_2x + a_2 \in [x_1, x_2] \\ .... \\ s_n(x) = d_nx^3 + c_nx^2 + b_nx + a_n \in [x_{n-1}, x_n] \end{cases} \quad (1)$$

where $(x_0, x_1, \cdots\cdots, x_n)$ is interpolation points; $(a_0, a_1, \cdots\cdots, a_n)$, $(b_0, b_1, \cdots\cdots, b_n)$, $(c_0, c_1, \cdots\cdots, c_n)$ and $(d_0, d_1, \cdots\cdots, d_n)$ are polynomial coefficients for the cubic spline interpolation piecewise function.

Condition 2: in the interval $\Delta = [a, b]$, $s(x)$ and its derivative function $s'(x)$ are continuous.

The free bending curve obtained by cubic spline interpolation is smooth.

The following theorem presents the parameter solution set characteristic of polynomial coefficients for the cubic spline interpolation piecewise function.

Theorem 1. *Assuming that the following conditions are given,* $s'_1(x_0) = B$, $s'_n(x_n) = B$, *there is only one solvable parameter solution set for polynomial coefficients for cubic spline interpolation piecewise function, and it can be mathematically expressed.*

Proof of Theorem 1. Assuming that $s(x_i) = y_i$.

Consider the following equalities, $s_i(x) = d_i(x_i - x_i)^3 + c_i(x_i - x_i)^2 + b_i(x_i - x_i) + a_i = y_i$, thus $a_i = y_i$.

In addition, define $h_i = x_{i+1} - x_i$, where h_i is the step length because $s_i(x_{i+1}) = y_i$. Moreover, $s'_i(x_{i+1}) = s'_{i+1}(x_{i+1})$, $s''_i(x_{i+1}) = s''_{i+1}(x_{i+1})$, make $m_i = s''_i(x_i) = 2c_i + 6d_i(x_i - x_i) = 2c_i$, and m_i is the introduced parameters, thus $b_i = \frac{y_{i+1}-y_i}{h_i} - \frac{h_i}{2}m_i - \frac{h_i}{6}(m_{i+1} - m_i)$, $c_i = \frac{m_i}{2}$, $d_i = \frac{m_{i+1}-m_i}{6h_i}$.

Make $\alpha_i = \frac{h_{i-1}}{h_{i-1}+h_i}$, $\beta_i = \frac{h_i}{h_{i-1}+h_i}$, due to $c_i = 6\left(\frac{y_{i-1}+y_i}{h_i} - \frac{y_{i-1}-y_i}{h_i}\right)\frac{1}{(h_{i-1}+h_i)}$, then $\alpha_i m_{i-1} + 2m_i + \beta_i m_{i-1} = c_i$, combined with the boundary conditions $s'_1(x_0) = B$, $s'_n(x_n) = B$. Introduced parameters m_i can be solved by the linear equations solving method. □

According to the above theorem, cubic spline interpolation is suitable for various industrial applications and has low computational complexity.

2.2. Automatic Parking Path Optimization Model Using Cubic Spline Interpolation

Automatic parking path optimization is a necessary calculation link for intelligent automatic parking. The intelligent automatic parking control system contains a parking optimization data collection device, track trajectory optimizer and its ancillary program, parking track controller, parking feasibility in determining device, emergency braker, parking braker, etc. [22]. The specific design structure diagram of automatic parking control system is shown in Figure 2.

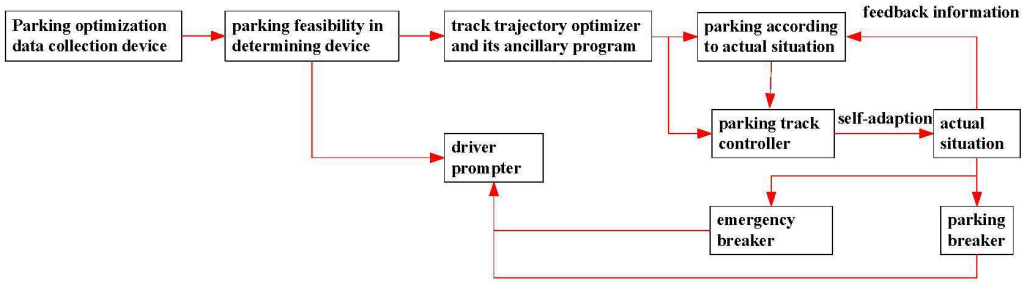


Figure 2. Design structure diagram of automatic parking control system.

According to the automatic parking path optimization model, the optimization objective is the shortest parking path, and the decision variables are several parking location reference points. There are four constraints: no collision between the vehicle and the garage side line during the parking process; the parking trajectory curve can be derived; the parking path is obtained by cubic spline interpolation; the parking location reference points are selected from the collected location reference candidate point set.

There are many forms for automatic parking; among them, reversing into a garage of automatic parking is a common parking scene. The specific schematic diagram of intelligent automatic parking is shown in Figure 3.

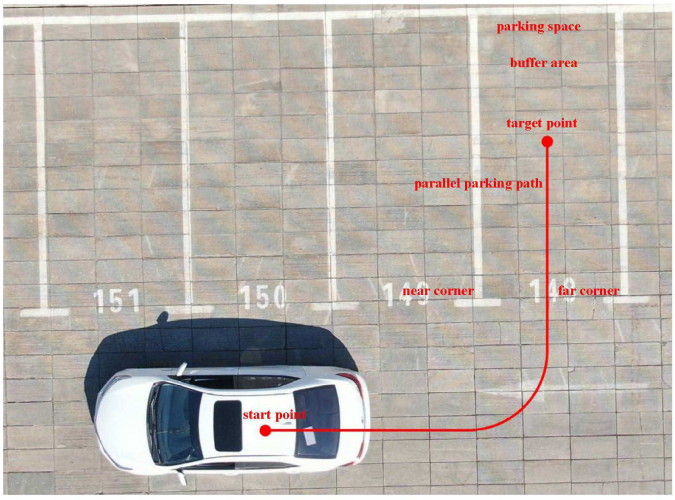


Figure 3. Schematic diagram of intelligent automatic parking.

As can be seen from Figure 3, the garage sideline and corner of the parking space are clear and identifiable, the vehicle is located in the start point initially, and it is necessary to park at the target point. Meanwhile, the automatic parking control system collects relative information immediately, and judges whether the parking conditions are met. On the premise of meeting parking conditions, according to the collected relative information, plan the optimal parking path and track precisely so as to park at the target point.

The specific automatic parking path optimization model is expressed as follows:

$$\min L = \int_0^{T_{\max}} v(t)dt \approx \sum_{it=1}^{n_t} \Delta L_{it}$$
$$S.T. \begin{cases} LT_{X,Y} = s(p_1, p_2, \ldots, p_{is}, \ldots, p_{n_s}) \\ \Delta L_{it} \le \Delta L_{\max} \& \Delta \angle_{it} \le \Delta \angle_{\max} \\ \forall p_{is} \in P_C \\ \forall P_g(i) \notin \Omega_{c,it} \\ \forall P_g(i) \in \Omega_g \end{cases} \tag{2}$$

where P_g is location set for the garage collision detection point; the number of set P_g is n_g; $P_g(i)$ is the location of the i-th garage collision detection point, $i \in \{1, 2, \ldots, n_g\}$; ΔL_{it} and $\Omega_{c,it}$ are the i-th time period parking interval distance and vehicle coverage area; the number of the distinct time period is n_t, $it \in \{1, 2, \ldots, n_t\}$, thus, the end time of parking is $T_{\max} = n_s \Delta t$, because the vehicles can't touch the garage edge, $\forall P_g(i) \notin \Omega_{c,it}$; L is the length of the parking path, and it is the definite integral parking velocity $v(t)$ about t from parking start 0 to parking end $T_{\max}$, approximately equal to the sum of the parking interval distances for all time periods; $\Delta L_{\max}$ and $\Delta \angle_{\max}$ are the maximum value of the parking interval distance and parking attitude angle for any allowed time period; $\Delta \angle_{it}$ are the i-th time period parking attitude angle, $\Delta L_{it} \le \Delta L_{\max} \& \Delta \angle_{it} \le \Delta \angle_{\max}$; $LT_{X,Y}$ is the parking trajectory in the plane coordinate system composed of the X axis and Y axis; $\{p_1, p_2, \ldots, p_{is}, \ldots, p_{n_s}\}$ is the parking location reference point set, $is \in \{1, 2, \ldots, n_s\}$; the number of set $\{p_1, p_2, \ldots, p_{is}, \ldots, p_{n_s}\}$ is n_s; $s(p_1, p_2, \ldots, p_{is}, \ldots, p_{n_s})$ is the cubic spline interpolation curve for set $\{p_1, p_2, \ldots, p_{is}, \ldots, p_{n_s}\}$, and it is the novel assumption; $PC = \{pc_1, pc_2, \ldots, pc_{ic}, \ldots, pc_{n_c}\}$ is the parking location reference candidate point set; the number of set PC is n_c, $\forall p_{is} \in PC$; and n_s is much smaller than n_c.

For the specific reversing into a garage of automatic parking, the following two constraints should be added. (I) The optimization effect should be compared under fair comparison conditions. In other words, when the vehicle is finally parked, the error between the distance between the center of the vehicle and the bottom line of the garage and the expected distance should be less than the allowable range. (II) At the final stage of parking, the vehicle body should be parallel to the side line of the garage as far as possible.

In other words, when the vehicle is finally parked, the error between the distance between the center of the vehicle and the midpoint of the vehicle bottom and 1/2 of the vehicle length should be less than the allowable range.

The specific schematic diagram of automatic parking path optimization using cubic spline interpolation is shown in Figure 4.

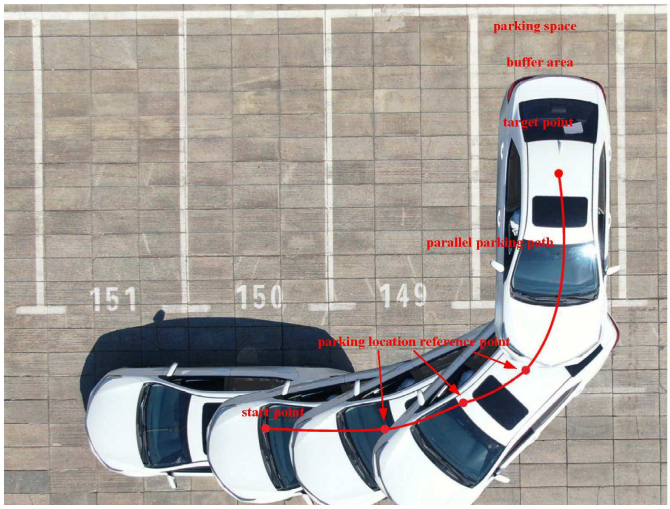


Figure 4. Schematic diagram of automatic parking path optimization using cubic spline interpolation.

Generally, the bottom garage corner, far away from vehicle, is set as the reference origin (0,0), and the length and position unit are m; fixed length binary coding is adopted, where the coding length is equal to the number of parking location reference point set n_s, and $X = TtoB(\{ic_1, ic_2, \ldots, ic_{is}, \ldots, ic_{n_s}\})$ is binary coding for the parking location reference point set, $TtoB(ic_{is})$ is the binary number of ic_{is}, and $pc_{ic_{is}}$ is the ic_{is}-th element of the parking location reference candidate point set PC.

3. Moth Flame Optimization Algorithm

The moth flame optimization algorithm (MFO) is an effective optimization algorithm, proposed by S. Mirijalili in 2015. The algorithm abstracts the optimization algorithm by observing the phenomenon that moths locate the light source horizontally [23]. In the moth flame optimization algorithm, the moth population position represents a series of candidate solution sets, and the flame position represents a series of elite solution sets currently.

The moth population position is represented by matrix M, and it is expressed as follows.

$$M = \begin{bmatrix} M_{11} & M_{12} & \ldots & M_{1d} \\ M_{21} & M_{22} & \ldots & M_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ M_{n1} & M_{n2} & \ldots & M_{nd} \end{bmatrix} \tag{3}$$

where n represents the number of moths; d represents the dimension of the solution.

The moth population fitness value is represented by matrix OM, and it is expressed as follows.

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{4}$$

Initially, the flame number is equal to the moth number, and the initial flame group position is represented by matrix F. It is expressed as follows.

$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1d} \\ F_{21} & F_{22} & \dots & F_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n1} & F_{n2} & \dots & F_{nd} \end{bmatrix} \tag{5}$$

The flame group fitness value is represented by matrix OF, and it is expressed as follows.

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{6}$$

The optimization process can be abstracted as a triplet.

$$MFO = (I, P, T) \tag{7}$$

where I means the initialization step: generate moth individuals randomly, and obtain their fitness value. The specific mapping relationship is expressed as follows.

$$I : \phi \rightarrow \{M, OM\} \tag{8}$$

where P indicates the position updating behavior: according to the logarithmic spiral law, moth individuals update their positions based on the information of themselves and the flame group [24]. If a better fitness value than the current one is obtained, the corresponding flame needs to be updated. The specific mapping relationship is expressed as follows.

$$P : M \rightarrow M \tag{9}$$

The specific position updating is expressed as follows.

$$\begin{cases} M_i = S(M_i, F_j) = D_{ij} \cdot e^{\tau t} \cdot \cos(2\pi t) + F_j \\ D_{ij} = |M_i - F_j| \end{cases} \tag{10}$$

where M_i represents the i-th moth position; F_j represents the j-th flame position; D_{ij} represents the Euclidean distance between the i-th moth and the j-th flame; τ represents the parameter for changing the logarithmic spiral waveform; t represents a random number between -1 and 1; and $S(M_i, F_j)$ represents a logarithmic spiral function.

In addition, aiming at accelerating the convergence speed, the flame number will be adaptively reduced with the increase in the number of iterations. The specific update mechanism for the flame number is expressed as follows.

$$N_{flame} = round(N_{flame}^{\max} - l\frac{N_{flame}^{\max} - 1}{N_{\text{int}}^{\max}}) \tag{11}$$

where N_{flame} is the updated flame number; $N_{flame}^{\max}$ is the maximum number of flames; l is the iteration current number; and $N_{\text{int}}^{\max}$ is the iteration maximum number.

4. Improved Immune Moth Flame Optimization Algorithm

4.1. Immune Mechanism

The essence of the immune mechanism is as follows: the global optimal solution to be found of the population is taken as the antigen, and each individual in the population is taken as the antibody. Once the antibody is stimulated by an external invasion, it constantly

produces new antibodies to realize immune function. However, if necessary control measures are lacking, the immune cells with a higher concentration will monopolize the whole population. The immune mechanism maintains the population diversity according to the concentration selection strategy so as to avoid the local convergence of the algorithm and improve its global optimization ability. If the immune mechanism based on the concentration selection strategy is introduced into the moth fire fighting algorithm, its optimization efficiency will be improved. In the concentration selection strategy, the calculation formula of the antibody concentration and its concentration probability is as follows.

$$D(x_i) = \frac{1}{\sum_{j=1}^{m+N} \left| f(x_i) - f(x_j) \right|} \tag{12}$$

$$P_d(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{i=1}^{m+N} \frac{1}{D(x_i)}} = \frac{\sum_{j=1}^{m+N} \left| f(x_i) - f(x_j) \right|}{\sum_{i=1}^{m+N} \sum_{j=1}^{m+N} \left| f(x_i) - f(x_j) \right|} \tag{13}$$

where $f(x_i)$, $D(x_i)$ and $P_d(x_i)$ are the fitness function value, antibody concentration and its concentration probability of each moth, $i = 1, 2, \ldots, m + N$.

In the concentration selection strategy, the immune system will promote the production of antibodies with great lethality to the antigen; on the contrary, it will inhibit antibodies with low lethality and high concentration. So, the balance between global search and local development can be effectively enhanced.

4.2. Gaussian Mutation for Flame Updating

Gaussian distribution is also called normal distribution. Gaussian disturbance is a kind of disturbance whose intensity conforms to Gaussian distribution (normal distribution). The mechanism of realizing mutation by applying Gaussian disturbance is called the Gaussian mutation mechanism. According to the characteristics of normal distribution, Gaussian mutation can realize the key search in the local area near the original individual. Therefore, the introduction of the Gaussian mutation mechanism has two important effects that cannot be ignored: I It can effectively expand and strengthen the local search range and intensity of the moth flame optimization algorithm so as to help improve its local development ability; II When the moth flame optimization algorithm has the risk of local convergence in a certain local area, the strong local disturbance in this area will significantly help it escape from local convergence [20]. Obviously, the introduction of the Gaussian mutation mechanism is an effective strategy to enhance the balance between the global development and local convergence of the moth flame algorithm. In this paper, a new flame position updating method based on the moth flame algorithm with Gauss mutation is adopted. For the flame position F_i with index i, the specific flame position update formula with Gaussian variation is as follows.

$$\begin{aligned} \lambda_{ik} &= \frac{\sum_{j=1}^{N} F_{jk}}{N} \\ V_{ik} &= F_{ik} + \lambda_{ik} \cdot Gaussian\left(\mu, \sigma^2\right) \end{aligned} \tag{14}$$

where F_{jk} represents the k-th dimension of the flame selected by the moth with index j, λ_{ik} represents the k-th dimension of the weight vector conforming to the flame disturbance characteristics of the entire moth population, F_{ik} and V_{ik} represent the original value of the k-th dimension of the flame position F_i with index i and the updated value after Gaussian mutation. $Gaussian\left(\mu, \sigma^2\right)$ is a Gaussian distribution (normal distribution) random number with mean μ and standard deviation σ.

4.3. OBL Mechanism for Elite

Opposition-based learning is abbreviated as OBL. The OBL mechanism is an effective mechanism to improve the optimization quality of the bionic optimization algorithm, and it was proposed by Tizhoosh in 2005 [25]. The OBL mechanism has the ability to generate a certain number of elite opposition-based learning solutions far away from the local optimum. If the moth flame algorithm is placed in a circumstance of falling into local convergence, these elite opposition-based learning solutions can make the population move away from the local region so as to promote the balance quality between the global search and local development capability.

The specific elite opposition-based learning formula is as follows:

$$\begin{cases} x(t)'_i = k(a_i + b_i) - x(t)_i \\ x(t)'_i, x(t)_i \in [a_i, b_i] \\ i \in [1, 2, \ldots\ldots, d] \end{cases} \tag{15}$$

where a_i and b_i represent the minimum and maximum values on the boundary of the i-th dimension; $k \in [0, 1]$ is the random generalization coefficient; $x(t)'_i$ and $x(t)_i$ represent the elite opposition-based learning and original solution of the i-th dimension; and d is the number of dimension.

However, there is a probability of 'overflow' for elite opposition-based learning in the iteration process. If the i-th dimension exceeds the boundary, the overflow solution is handled by the overflow strategy.

The specific overflow disposal formula is as follows:

$$x(t)'_i = a_i + \beta(b_i - x(t)_i) \tag{16}$$

where $\beta \in [0, 1]$ is the random overflow disposal coefficient.

4.4. Adaptive Decreasing Inertia Weight Coefficient

Since the inertia weight coefficient has a certain impact on the optimization effect of the algorithm, in order to improve the global search ability of MFO in the initial stage of iteration, it is necessary to select a larger inertia weight coefficient. On the contrary, in the late iteration, in order to enhance the local search ability, a smaller inertia weight coefficient should be selected. Obviously, introducing the adaptive decreasing inertia weight coefficient is conducive to balance maintenance for global search and local development. This paper proposes to introduce the adaptive decreasing inertia weight coefficient into the position updating behavior [26]. The calculation formula of the adaptive decreasing inertia weight coefficient is expressed as follows.

$$\varpi = \omega_{\max} - \omega_d * \left(\frac{l}{N_{\text{int}}^{\max}} \right)^{\alpha} \tag{17}$$

where α is the decreasing decline rate. It makes the decreasing rate in the whole iteration process have a significant difference. When $\alpha = 1$, then it is equivalent to linear decline.

The moth update formula using the adaptive decreasing inertia weight coefficient is expressed as follows.

$$\begin{cases} M_i = S(M_i, F_j) = D_{ij} \cdot e^{\tau t} \cdot \cos(2\pi t) + \varpi \cdot F_j \\ D_{ij} = |M_i - F_j| \end{cases} \tag{18}$$

4.5. Design of Improved Immune Moth Flame Optimization Algorithm

The design highlights of the improved automatic parking path optimization based on the improved immune moth flame algorithm (IIMFO) is the balance maintenance mechanism for global search and local development. Aiming at improving the balance quality, the OBL strategy, immune mechanism, and adaptive decreasing inertia weight

coefficient are integrated into the moth flame algorithm.The flowchart for the improved immune moth flame algorithm is shown in Figure 5.

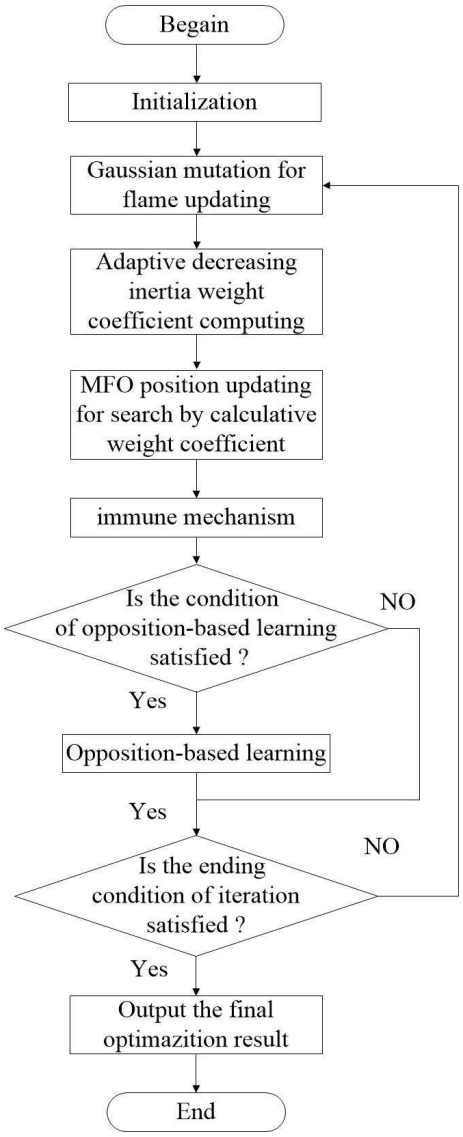


Figure 5. The flowchart for improved immune moth flame algorithm.

5. Simulation and Semi-Automatic Experiment

5.1. Test Function Simulation

In order to verify the effectiveness of the improved algorithm in this paper, three test functions (single objective function: De jong, Schaffer) simulated and compared the fourdifferent optimization algorithms: the improved immune moth flame optimization algorithm (IIMFO) proposed in this paper, the improved version of the moth flame optimization algorithm based on Lévy-flight strategy (LMFO), the moth flame optimization algorithm (MFO), differential evolution (DE), and particle swarm optimization (PSO).

The specific simulation comparison results are as follows:

(I) The sphere function (real optimal minimum function value is 0.0) is expressed as follows. $F(x_1, x_2) = {x_1}^2 + {x_2}^2$.

The specific simulation results of the sphere function are shown in Figure 6 and Table 1.

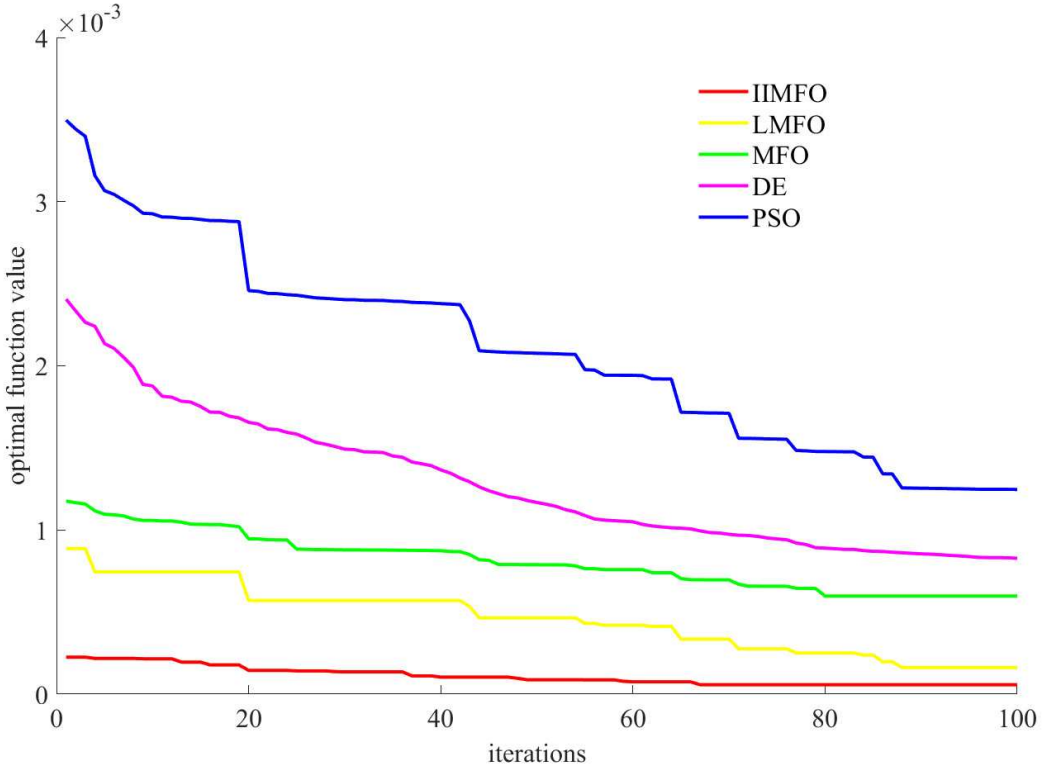


Figure 6. The convergence curves of sphere function.

Table 1. Optimization results of sphere function.

Algorithm	(x_1, x_2)	$F(x_1, x_2)$
IIMFO	$4.6 \times 10^{-3}, 6.0 \times 10^{-3}$	5.7×10^{-5}
IIMFO	$7.9 \times 10^{-3}, 9.8 \times 10^{-3}$	1.6×10^{-4}
MFO	$9.1 \times 10^{-3}, 2.3 \times 10^{-2}$	6.1×10^{-4}
DE	$1.6 \times 10^{-2}, 2.4 \times 10^{-2}$	8.3×10^{-4}
PSO	$2.1 \times 10^{-2}, 2.7 \times 10^{-6}$	1.2×10^{-3}

As shown in Figure 6 and Table 1, compared with LMFO, MFO, DE and PSO, the improved immune moth flame optimization algorithm is most ideal, and its optimal minimum function value is $(x_1, x_2) = (4.6 \times 10^{-3}, 6.0 \times 10^{-3})$, the optimal function value is $F(x_1, x_2) = 5.7 \times 10^{-5}$.

(II) De jong function (real optimal minimum function value is 0.0), it is expressed as follow. $F(x_1, x_2) = 100 * (x_1 - x_2)^2 + (1 - x_1)^2$.

The specific simulation results about De jong function are shown in Figure 7 and Table 2.

Table 2. Optimization results about De jong function.

Algorithm	(x_1, x_2)	$F(x_1, x_2)$
IIMFO	0.9929, 0.9933	6.7×10^{-5}
LMFO	0.9922, 0.9938	3.1×10^{-4}
MFO	0.9796, 0.9809	5.9×10^{-4}
DE	0.9765, 0.9789	1.1×10^{-3}
PSO	0.9774, 0.9802	1.3×10^{-3}

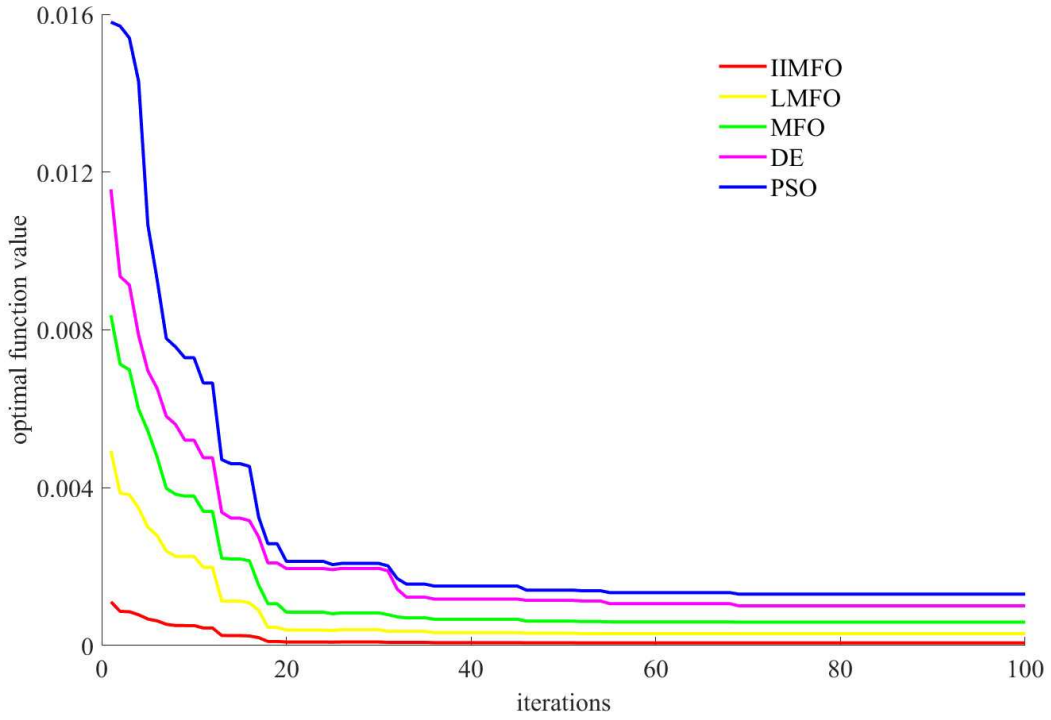


Figure 7. The convergence curves about De jong function.

As shown in Figure 7 and Table 2, compared with LMFO, MFO, DE and PSO, the improved immune moth flame optimization algorithm is most ideal, and its optimal minimum function value is $(x_1, x_2) = (0.9929, 0.9933)$, the optimal function value is $F(x_1, x_2) = 6.7 \times 10^{-5}$.

(III) The Schaffer function (real optimal minimum function value is 0.0) is expressed as follows. $F(x_1, x_2) = 0.5 + \frac{((\sin({x_1}^2+{x_2}^2)^{0.5})-0.5)}{(1+0.001\times({x_1}^2+{x_2}^2))^2}$.

The specific simulation results of the Schaffer function are shown in Figure 8 and Table 3.

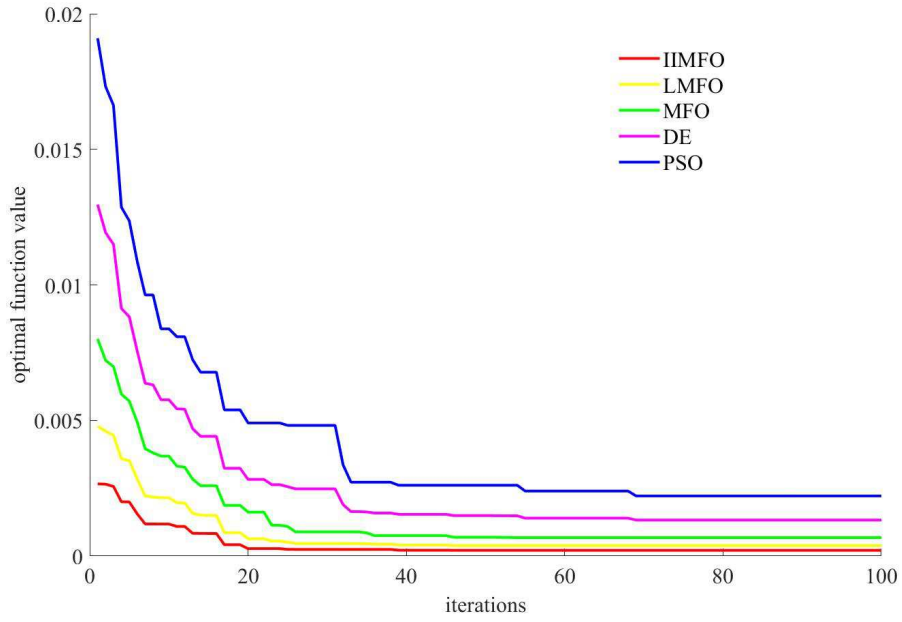


Figure 8. The convergence curves of Schaffer function.

Table 3. Optimization results of Schaffer function.

Algorithm	(x_1, x_2)	$F(x_1, x_2)$
IIMFO	8.7×10^{-5}, 9.1×10^{-5}	1.3×10^{-4}
LMFO	2.9×10^{-4}, 2.5×10^{-4}	3.8×10^{-4}
MFO	5.5×10^{-4}, 4.0×10^{-4}	6.8×10^{-4}
DE	8.9×10^{-4}, 1.0×10^{-3}	1.4×10^{-3}
PSO	1.6×10^{-3}, 1.3×10^{-3}	2.1×10^{-3}

As shown in Figure 8 and Table 3, compared with LMFO, MFO, DE and PSO, the improved immune moth flame optimization algorithm is most ideal, and its optimal minimum function value is $(x_1, x_2) = (8.7 \times 10^{-5}, 9.1 \times 10^{-5})$, the optimal function value is $F(x_1, x_2) = 1.3 \times 10^{-4}$.

5.2. Intelligent Automatic Parking Simulation

The setting situation of coordinate axis for intelligent automatic parking in this paper is as follows: the bottom edge line of the vehicle garage is the x-axis, the side line of the vehicle garage away from the vehicle and vertical to the x-axis is the y-axis, the bottom corner of garage edge away from the vehicle is the reference origin O, and the coordinate of reference origin O is (0,0). The two automatic parking scenarios are chosen: Volkswagen UP and Honda XR-V are the simulation object, and the parallel distance between the initial coverage area and vehicle garage near the corner is 2.2 m, respectively. The specific setting situation of scenario for Volkswagen UP is described as follows: the parking space is 5.0×2.5 m^2, the vehicle coverage area is 3.5×1.7 m^2, and the distance between initial coverage area and the side line of vehicle garage is 1 m. The scenario for Honda XR-V is as follows: the parking space is 5.0×2.5 m^2, the vehicle coverage area is 4.4×1.8 m^2, and the distance between initial coverage area and side line of vehicle garage is 0.8 m.

The parking path optimization method adopts the improved immune moth flame optimization algorithm proposed in this paper (IIMFO), improved version of the moth flame optimization algorithm based on Lévy-flight strategy (LMFO), the moth flame optimization algorithm (MFO), differential evolution (DE) and particle swarm optimization (PSO). The specific automatic parking path curves and optimization results using Volkswagen UP and Honda XR-V are shown in Figures 9 and 10 and Tables 4 and 5.

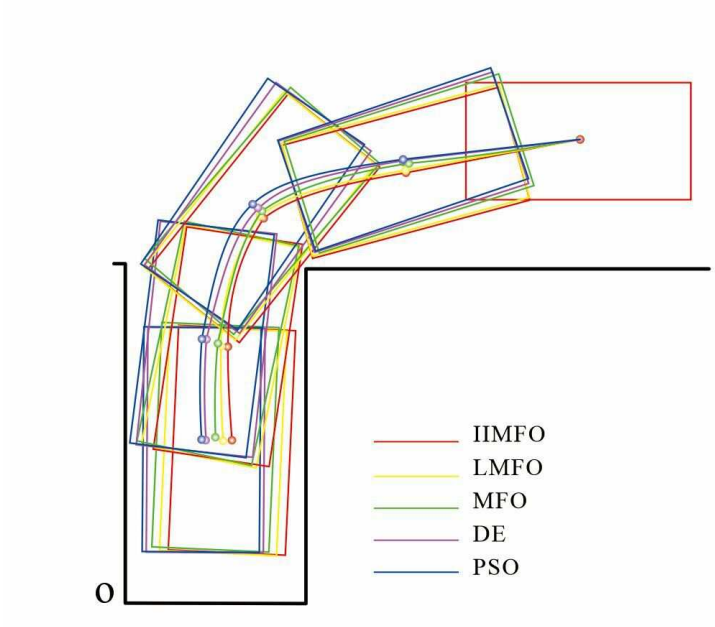


Figure 9. Automatic parking path curves using Volkswagen UP.

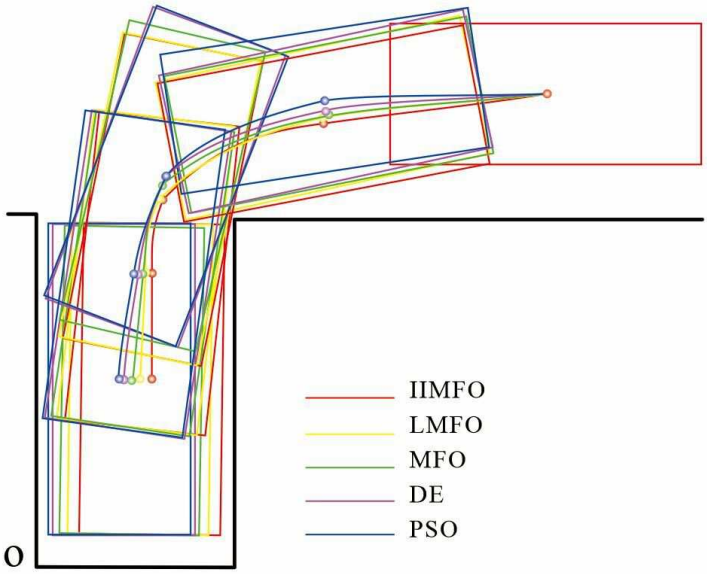


Figure 10. Automatic parking path curves using Honda XR-V.

Table 4. Optimization results about automatic parking using Volkswagen UP.

Algorithm	Parking Location Reference Point Set (m)	Path Length (m)
IIMFO	(1.49, 2.37), (1.41, 3.82), (1.98, 5.73), (4.06, 6.47)	9.27
LMFO	(1.30, 2.40), (1.32, 3.80), (1.96, 5.75), (4.05, 6.48)	9.41
MFO	(1.20, 2.49), (1.29, 3.85), (1.95, 5.82), (4.12, 6.62)	9.75
DE	(1.06, 2.44), (1.09, 3.93), (1.91, 5.84), (3.98, 6.66)	10.27
PSO	(1.03, 2.45), (1.07, 3.93), (1.86, 5.87), (3.98, 6.67)	10.35

Table 5. Optimization results about automatic parking using Honda XR-V.

Algorithm	Parking Location Reference Point Set (m)	Path Length (m)
IIMFO	(1.56, 2.21), (1.58, 4.23), (1.64, 5.78), (3.88, 6.42)	9.14
LMFO	(1.36, 2.22), (1.50, 4.24), (1.63, 5.79), (3.86, 6.45)	9.33
MFO	(1.26, 2.19), (1.46, 4.22), (1.67, 5.91), (3.95, 6.52)	9.72
DE	(1.16, 2.19), (1.40, 4.20), (1.70, 5.96), (3.90, 6.54)	10.07
PSO	(1.11, 2.20), (1.36, 4.21), (1.70, 5.96), (3.90, 6.65)	10.13

As shown in the automatic parking path curves and related results using Volkswagen UP and Honda XR-V (Figures 9 and 10 and Tables 4 and 5), compared with MFO, DE and PSO, under the premise of no collision avoidance of the side line of the vehicle garage during the parking process, the automatic parking path obtained by IIMFO is smoother and shorter. This indicates that IIMFO proposed in this paper has a stronger optimization effect and has more advantages in intelligent parking.

5.3. Intelligent Automatic Parking Semi-Automatic Experiment

Because the ideal automatic experiment environment is not easy to be obtained, a semi-automatic experiment environment is chosen as the verification environment in this paper. A commonly used and favored intelligent semi-automatic parking mode was adopted in the semi-automatic experiment environment. The specific details about the limitations of the intelligent automatic parking semi-automatic experiment are shown as follows.

(I) The parking system contains the identification function of extremely bad weather and road conditions. When it knows that the vehicle is in extremely bad weather and under different road conditions such as rainstorm, blizzard, heavy uphill and heavy downhill, it will automatically disable the intelligent semi-automatic parking function and inform the driver.

(II) The parking system is equipped with both a parking path optimization function and parking path tracking control performance. In other words, in general, parking systems have the potential to realize unmanned autonomous parking.

(III) In the process of intelligent semi-automatic parking, it is necessary to configure a driver familiar with two simple skills. The details are as follows: I according to the deviation between the optimized path and the tracking control path in the reversing image, manually and gently control the steering wheel to improve the accuracy of the parking path tracking control; II it is able to stop parking or apply emergency braking in combination with the on-site parking situation and the prompt of the parking reminder. In other words, the driver acts as both a safety officer and an enhanced track control corrector.

(IV) The parking system is equipped with strong safety assurance measures. Once an accident occurs, the parking vehicle will be parked immediately to ensure the safety of people and vehicles.

The above intelligent semi-automatic parking mode introduced in this paper will become a major parking mode in China in the future because of its dual advantages of convenience and economy.

In the above semi-automatic parking mode, the rearview camera with an optimization parking path and track trajectory is used for reference, and a driver with the above described simple skills is needed. Specifically, the optimization parking path is the capital driving factor, and the driving experience should not be ignored too. The driving trajectory tracks the optimization parking path by the rearview camera and corrects by the driving experience. So, the experiment vehicle need to configure the rearview camera with the optimization parking path and tracking trajectory, position sensors, driver prompter, tracking controller, parking feasibility decider, emergency parking device and stopping parking device.

In this paper, the No. 148 and No. 146 parking areas of Xinghai Square Shell Museum in Dalian, China are chosen as the semi-automatic experiment area, and the Toyota LeiLing Shuangqing 185T Sportline and First Automobile Works (FAW) Senya r7 are chosen as the semi-automatic experiment object. The two automatic parking scenarios are chosen: Toyota LeiLing Shuangqing 185T Sportline and Honda Accord 15T as semi-automatic experiment objects, and the parallel distance between initial coverage area and vehicle garage near corner is 2.2 m, respectively. The specific setting situation of scenario for FAW Senya r7 is described as follows: the parking space is 5.0×2.5 m^2, the vehicle coverage area is 4.7×1.9 m^2, and the distance between initial coverage area and the side line of vehicle garage is 1 m. The specific setting situation of scenario for Honda Accord 15T is described as follows: the parking space is 5.0×2.5 m^2, the vehicle coverage area is 4.4×1.8 m^2, and the distance between initial coverage area and the side line of vehicle garage is 1.2 m. The main parameters of the calculus complexity for semi-automatic experiments based on intelligent automatic parking in this paper are as follows: the limit time for collecting parking path optimization data is 4 s, the time for determining the feasibility of simulated parking is 1.5 s, the time for obtaining the parking path by the optimization algorithms is 10.5 s, and the limit of the total time for the parking path optimization is 16 s. In general, the acceptable limit total time is identified as 20 s, thus, the verification for semi-automatic experiment comparison in this paper is believable and realistic. The integral fitting method is used to solve the length of the stopping trajectory curve in the control period of 1000 μs (1 ms). In this paper, the Mavic 2 DJI's unmanned aerial vehicle (UAV) is chosen as the camera equipment. The specific physical diagram of Mavic 2 DJI's unmanned aerial vehicle is shown in Figure 11.



Figure 11. Physical diagram of Mavic 2 DJI's unmanned aerial vehicle.

In this paper, semi-automatic experiment comparison based on intelligent automatic parking was implemented on several calm, cloudless days. The parking path optimization method adopts the improved immune moth flame optimization algorithm proposed in this paper (IIMFO), improved version of moth flame optimization algorithm based on Lévy-flight strategy (LMFO), moth flame optimization algorithm (MFO), differential evolution (DE) and particle swarm optimization (PSO). The three vital fixed points were filmed using the Mavic 2 DJI's unmanned aerial vehicle. The specific filmed vital fixed points for the semi-automatic parking process using Volkswagen Toyota LeiLing Shuangqing 185T Sportline and FAW Senya r7 are shown in Figures 12 and 13.

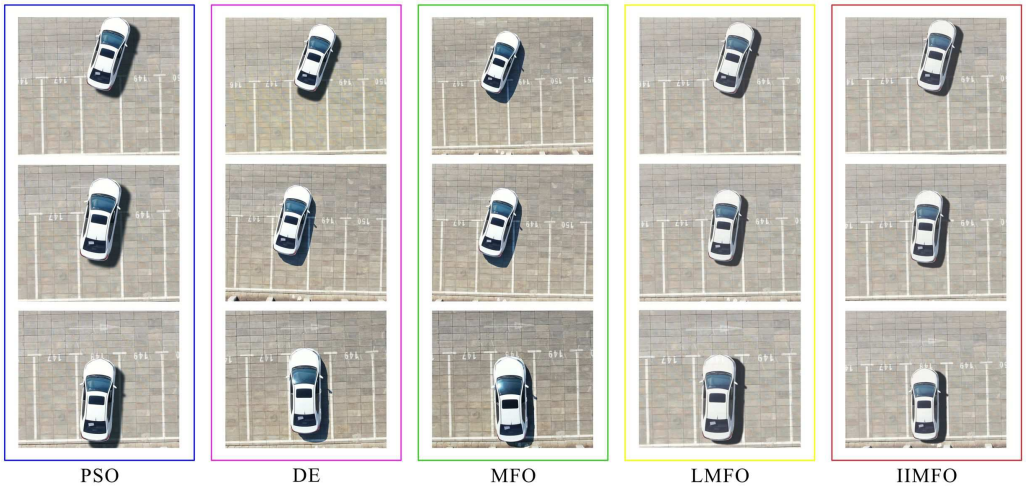


Figure 12. Filmed vital fixed points for automatic parking process using Volkswagen Toyota LeiLing Shuangqing 185T Sportline.

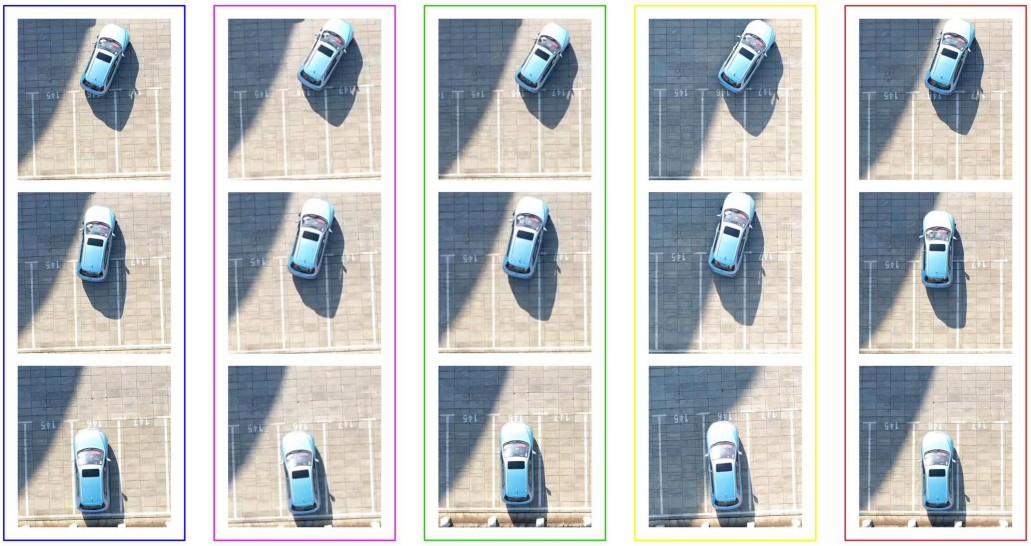


Figure 13. Filmed vital fixed points for automatic parking process using FAW Senya r7.

As can be seen from the filmed vital fixed points for the semi-automatic parking process using Toyota LeiLing Shuangqing 185T Sportline and FAW Senya r7, compared with LMFO, MFO, DE and PSO, under the premise of no collision avoidance of the side line of the vehicle garage during the parking process, the semi-automatic parking effect obtained by IIMFO is more ideal. It indicates that the IIMFO algorithm is a suitable algorithm with strong optimization ability and can deal with the actual automatic parking problem more effectively.

6. Conclusions and Future Work

In view of the disadvantages of the optimization method in the traditional automatic parking control system, this paper constructs an automatic parking path optimization model based on cubic spline interpolation and proposes an improved immune moth flame algorithm for automatic parking path optimization. The following summarizes the main innovation of the paper:

(I) Innovations of automatic parking path optimization method: aiming at the problem that the optimal automatic parking path chosen by the optimization method is not ideal for path length and the tracking trajectory is not smooth enough, an automatic parking path optimization model based on cubic spline interpolation is constructed.

(II) Innovations of the MFO algorithm: aiming at the problems that the automatic parking control system needs to optimize the parking path and that the efficiency of the traditional optimization algorithm is low, this paper proposes to use MFO to optimize the parking path, and to solve the defect that MFO lacks the effective mechanism of balance between global search and local development, four improvements are proposed. Four specific novel strategies are as follows: I introduce a immune mechanism in the iterative process, which can adjust the distribution of moths in space so as to jump out of the local optimization and then seek the global optimization; II introduce the Gaussian mutation mechanism to provide strong local distribution so as to explore the key local space deeply enough and enhance the ability to jump out of local convergence; III introduce the adaptive decreasing inertia weight coefficient to further strengthen the global optimization capabilities; IV introduce the OBL mechanism to further improve the optimization quality.

The numerical results of the test function show that the improved strategies are excellent for improving the optimization precision, iterative accuracy and stability of the MFO. Further, the results of simulation and semi-automatic experiment comparison based

on intelligent automatic parking indicate that IIMFO has a better, improved optimization effect than was anticipated.

There are several suggestions as future directions for automatic parking path optimization based on the immune moth flame algorithm: I draw lessons from the existing improved MFO, such as EMFO, B-MFO or MTV-MFO, to further improve its optimization performance; II designing a novel automatic parking path optimization model by using other types of spline interpolation, such as cubic B-spline or high cardinal spline, so as to improve computational efficiency and accuracy; III enhance the MFO algorithm by integrating into other mechanisms or methods so as to further improve the optimization quality; IV combine the improved MFO with other algorithms, such as the differential evolution (DE) algorithm or genetic algorithm (GA), to further improve its optimization performance; V construct a verification environment using the automatic parking path experiment and the quantitative method so as to further improve the verification precision.

Author Contributions: The work presented here was performed in collaboration among all authors. Y.C. designed, analyzed, and wrote the paper and completed the simulation and experiment. L.W. provided funds and guided this paper. G.L. conceived idea and designed optimization algorithm. B.X. involved the simulation and experiment. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by key projects of natural science research in Anhui Universities grant number KJ2019A0860, National Science Fund cultivation project of the Inner Mongolia University for Nationalities grant number NMDGP17101, Inner Mongolia University for Nationalities doctoral research initiation Fund Project grant number BS416.

Data Availability Statement: Not applicable.

Acknowledgments: This work was financially supported by key projects of natural science research in Anhui Universities (KJ2019A0860), National Science Fund cultivation project of the Inner Mongolia University for Nationalities (NMDGP17101), and Inner Mongolia University for Nationalities doctoral research initiation Fund Project (BS416).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IIMFO	improved immune moth flame optimization
LMFO	moth flame optimization algorithm based on Lévy-flight strategy
MFO	moth flame optimization
B-MFO	binary moth-flame optimization
MTV-MFO	multi-trial vector-based moth-flame optimization
EMFO	enhanced moth-flame optimization
DE	differential evolution
PSO	particle swarm optimization
GA	genetic algorithm

References

1. Zhang, P.; Xiong, L.; Yu, Z.; Fang, P.; Yan, S.; Yao, J.; Zhou, Y. Reinforcement Learning-Based End-to-End Parking for Automatic Parking System. *Senors* **2019**, *19*, 3996. [CrossRef] [PubMed]
2. Liu, K.; Dao, M.; Inoue, T. An Exponentially *epsilon*-Convergent Control Algorithm for Chained Systems and Its Application to Automatic Parking Systems. *IEee Trans. Control Syst. Technol.* **2006**, *14*, 1113–1126. [CrossRef]
3. Liu, H.; Luo, S.; Lu, J. Method for Adaptive Robust Four-Wheel Localization and Application in Automatic Parking Systems. *IEEE Sens. J.* **2019**, *19*, 10644–10653. [CrossRef]
4. Suhr, J.K.; Jung, H.G. Automatic Parking Space Detection and Tracking for Underground and Indoor Environments. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5687–5698. [CrossRef]
5. Chai, R.; Tsourdos, A.; Savvaris, A.; Chai, H.; Xia, Y.; Chen, C.L.P. Design and Implementation of Deep Neural Network-Based Control for Automatic Parking Maneuver Process. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1400–1413. [CrossRef]

6. Kawecki, L.; Niewierowicz, T. Hybrid Genetic Algorithm to Solve the Two Point Boundary Value Problem in the Optimal Control of Induction Motors. *IEEE Lat. Am. Trans.* **2014**, *12*, 176–181. [CrossRef]
7. Islam, S.M.; Das, S.; Ghosh, S.; Roy, S.; Suganthan, P.N. An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Trans. Syst. Man Cybern. Part (Cybern.)* **2012**, *42*, 482–500. [CrossRef] [PubMed]
8. Gaurav, D.; Krishna, K.; Adam, S.; Victor, C.; Meenakshi, G. EMoSOA: A new evolutionary multi-objective seagull optimization algorithm for global optimization. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 571–596.
9. Kumar, S.; Tejani, G.G.; Mirjalili, S. Modified symbiotic organisms search for structural optimization. *Eng. Comput.* **2019**, *35*, 1269–1296. [CrossRef]
10. Tejani, G.G.; Kumar, S.; Gandomi, A.H. Multi-objective heat transfer search algorithm for truss optimization. *Eng. Comput.* **2021**, *37*, 641–662. [CrossRef]
11. Wang, X.; Shi, H.; Zhang, C. Path Planning for Intelligent Parking System Based on Improved Ant Colony Optimization. *IEEE Access* **2020**, *8*, 65267–65273. [CrossRef]
12. Aziz, M.A.E.; Ewees, A.A.; Hassanien, A.E. Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Syst. Appl.* **2017**, *83*, 242–256. [CrossRef]
13. Wang, M.; Chen, H.; Yang, B.; Zhao, X.; Hu, L.; Cai, Z.; Huang, H.; Tong, C. Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* **2017**, *267*, 69–84. [CrossRef]
14. Zhang, B.; Tan, R.; Lin, C. Forecasting of e-commerce transaction volume using a hybrid of extreme learning machine and improved moth-flame optimization algorithm. *Appl. Intell.* **2021**, *51*, 952–965. [CrossRef]
15. Zhao, X.; Fang, Y.; Liu, L.; Li, J.; Xu, M. An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems. *Appl. Intell.* **2020**, *50*, 4434–4458. [CrossRef]
16. Li, Y.; Zhu, X.; Liu, J. An Improved Moth-Flame Optimization Algorithm for Engineering Problems. *Symmetry* **2020**, *12*, 1234. [CrossRef]
17. Nadimi-Shahraki, M.H.; Banaie-Dezfouli, M.; Zamani, H.; Taghian, S.; Mirjalili, S. B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets. *Computers* **2021**, *10*, 136. [CrossRef]
18. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Ewees, A.A.; Abualigah, L.; Abd Elaziz, M. MTV-MFO: Multi-Trial Vector-Based Moth-Flame Optimization Algorithm. *Symmetry* **2021**, *13*, 2388. [CrossRef]
19. Li, Z.; Zhou, Y.; Zhang, S.; Song, J. Lévy-fight Moth-fame algorithm for function optimization and engineering design problems. *Math. Probl. Eng.* **2016**, *2016*, 1423930. [CrossRef]
20. Xu, L.; Li, Y.; Li, K.; Beng, G.H.; Jiang, Z.; Wang, C.; Liu, N. Enhanced Moth-flame Optimization Based on Cultural Learning and Gaussian Mutation. *J. Bionic Eng.* **2018**, *15*, 751–763. [CrossRef]
21. Dyer, S.A.; Dyer, J.S. Cubic-spline interpolation. 1. *IEEE Instrum. Meas. Mag.* **2001**, *4*, 44–46. [CrossRef]
22. Yan, C.; Yewang, Q. Improved Particle Swarm Optimization Algorithm for Automatic Entering Parking Space Based on Spline Theory. In Proceedings of the 2nd International Conference on Computing and Data Science (CONF-CDS 2021), Stanford, CA, USA, 28–30 January 2021; Volume 28, pp. 136–143.
23. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]
24. Li, C.; Li, S.; Liu, Y. A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting. *Appl. Intell.* **2016**, *45*, 1166–1178. [CrossRef]
25. Tizhoosh, H.R. Opposition-Based Learning: A New Scheme for Machine Intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control & Automation, & International Conference on Intelligent Agents, Web Technologies & Internet Commerce, Vienna, Austria, 28–30 November 2005; pp. 695–701.
26. Parsopoulos, K.E.; Vrahatis, M.N. Recent approaches to global optimization problems through Particle Swarm Optimization. *Nat. Comput.* **2002**, *1*, 235–306. [CrossRef]