

Article

Modified Mayfly Algorithm for UAV Path Planning

Xing Wang ¹, Jeng-Shyang Pan ¹, Qingyong Yang ¹, Lingping Kong ², Václav Snášel ²
and Shu-Chuan Chu ^{1,*}

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; wang_sing@sdust.edu.cn (X.W.); jspan@cc.kuas.edu.tw (J.-S.P.); yang_qy@sdust.edu.cn (Q.Y.)

² Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 70032 Ostrava, Czech Republic; lingping.kong@vsb.cz (L.K.); vaclav.snasel@vsb.cz (V.S.)

* Correspondence: scchu0803@sdust.edu.cn

Abstract: The unmanned aerial vehicle (UAV) path planning problem is primarily concerned with avoiding collision with obstacles while determining the best flight path to the target position. This paper first establishes a cost function to transform the UAV route planning issue into an optimization issue that meets the UAV's feasible path requirements and path safety constraints. Then, this paper introduces a modified Mayfly Algorithm (modMA), which employs an exponent decreasing inertia weight (EDIW) strategy, adaptive Cauchy mutation, and an enhanced crossover operator to effectively search the UAV configuration space and discover the path with the lowest overall cost. Finally, the proposed modMA is evaluated on 26 benchmark functions as well as the UAV route planning problem, and the results demonstrate that it outperforms the other compared algorithms.

Keywords: path planning; modified mayfly algorithm; exponent decreasing inertia weight; adaptive Cauchy mutation; enhanced crossover operator



Citation: Wang, X.; Pan, J.-S.; Yang, Q.; Kong, L.; Snášel, V.; Chu, S.-C. Modified Mayfly Algorithm for UAV Path Planning. *Drones* **2022**, *6*, 134. <https://doi.org/10.3390/drones6050134>

Academic Editor: Oleg Yakimenko

Received: 19 April 2022

Accepted: 18 May 2022

Published: 23 May 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

Unmanned aerial vehicles (UAV), as opposed to manned aircraft, refer to unmanned aircraft that can be controlled by remote radio control equipment or airborne computers. UAVs are less expensive and more flexible than manned aircraft, and they can reach difficult or dangerous places for humans to perform tasks. Therefore, UAVs have been widely used in civil and military fields. In the civil field, drones have played an important role in agricultural plant protection environment, aerial photography, express transportation, remote sensing of the oceans, and so on. In the military field, military drones can replace pilots in complex and dangerous missions such as intelligence gathering and reconnaissance surveillance. If the UAV is to complete the above tasks, one of the basic capabilities that the UAV must have is path planning capability [1].

The UAV path planning problem is to plan a collision-free path for the UAV from the starting point to the target point under the given flight conditions and flight environment. The planned path has the minimum cost and satisfies the relevant constraints. This problem can be simply described as an optimization problem with multiple constraints [2], so it is a challenge to traditional optimization strategies. There are already many mature solutions to this problem. In the direction of path planning based on graph search, Ref. [3] uses the A* algorithm for the UAV path planning problem with the goal of taking the least risk and consuming the least amount of fuel. Ref. [4] uses the D* Lite algorithm to plan an efficient path to the fire point for firefighting UAVs. In the direction of path planning based on sampling, Ref. [5] uses a hybrid algorithm combined with the artificial potential field method and the RRT-Connect algorithm for the path planning problem of UAV. Even though the techniques described above are relatively mature in mathematical theory, they are inefficient when working with discontinuous and non-derivative functions [6]. When

tackling the UAV route planning problem with many restrictions [7], they are readily trapped into local optimum solutions.

The path planning issue has been demonstrated to be an NP-hard problem, and the problem complexity grows with problem size [8]. When solving NP-hard problems, heuristic algorithms can produce high-quality solutions and are simple to implement. As a result, some well-developed meta-heuristics with good performance have been proposed in recent years, such as the Cuckoo Search algorithm (CS) [9,10], Particle Swarm Optimization (PSO) [11–13], Genetic Algorithm (GA) [14,15], Artificial Bee Colony (ABC) algorithm [16,17], Differential Evolution (DE) algorithm [18,19], Grey Wolf Optimizer (GWO) [20,21], Ant Colony Optimization (ACO) algorithm [22,23], etc.

The Mayfly algorithm (MA) [24] is a recently suggested optimization method, which takes the flight and mating behavior of the mayfly as the model. Female and male mayflies make up the entire mayfly population. The movements of the male and female mayflies offer the MA the capacity to conduct local searches, and the procedure of developing offspring through mayfly mating gives the MA global search capability.

At present, research on MA is still relatively small. Inspired by the bare-bones PSO (BBPSO) [25], and then, Juan Zhao [26] proposed the bare-bones mayfly algorithm. The chaotic mayfly algorithm was suggested by Mohamed A. M. Shaheen [27], who used a logical chaotic map to initialize the mayfly population. This paper presented a modified Mayfly Algorithm (modMA), which includes the exponent decreasing inertia weight strategy, adaptive Cauchy mutation, and an enhanced crossover operator to analyze the path planning issue of UAV from the ground station to the destination, and it achieves good results.

Many works on improving two-dimensional paths using metaheuristics have been proposed to this point. Ref. [28] used the bat and cuckoo algorithm in path planning. Ref. [29] used the grey wolf optimizer to tackle the UAV route planning issue. Ref. [30] used a chaotic cuckoo search algorithm to optimize drone paths in two-dimensional (2D) space. Besides, many researchers used other meta-heuristic algorithms to study UAV path planning problems [31–33]. However, no one has proposed the use of MA to tackle UAV path planning in a 2D environment. Hence, this paper will use the modified Mayfly Algorithm (modMA) to study this problem. The following are the contributions to this paper.

- (1) It proposes an enhanced crossover operator based on the MA, which can improve MA's exploration capability and convergence speed. Then it proposes a modified Mayfly Algorithm (modMA), which combines the EDIW strategy, adaptive Cauchy mutation, and the enhanced crossover operator to balance the MA's processes of exploration and exploitation;
- (2) It introduces three versions of the modMA and compares their performance on six 100 and 300 dimension benchmark functions;
- (3) Compare the three versions of the modified MA with other algorithms (MA, PSO, GWO, BOA) on twenty-six 50 dimension benchmark functions, and analyze the results of the experiments;
- (4) Use the three versions of the modified MA for tackling the 2D path planning problem of agricultural UAV.

This paper is organized as follows: The standard MA, as well as the problem of UAV path planning, are briefly described in Section 2. Section 3 provides three optimization ideas for improving the MA. In Section 4, two experiments are planned to compare the MA's enhanced performance using different optimization ideas and apply the three versions of the modified MA to the path planning problem of the agricultural UAV. Section 5 concludes.

2. Related Work

This section will provide a quick overview of the standard MA and the two-dimensional path planning problem of UAV.

2.1. Standard Mayfly Algorithm

The proposed MA’s fundamental concept is derived from the flight and mating behavior of mayflies. It combines the advantages of PSO [11], GA [14] and FA [34]. MA initially produces a mayfly population comprising males and females at random. The current velocity and position of the i th mayfly are two n -dimensional vectors, they are denoted as $v_i = (v_{i1}, v_{i2} \dots, v_{in})$ and $x_i = (x_{i1}, x_{i2} \dots, x_{in})$, respectively. Each mayfly modifies its position depending on its individual best ($pbest$) position and the best ($gbest$) position found by the whole mayfly swarm thus far.

2.1.1. Male Mayfly Flight

Male mayflies cluster together, which suggests that their positions are updated based on social and personal experience. x_i^t is defined as the present position of the i th male mayfly at iteration t . Assuming that x_i^t is the current position of the i th mayfly at iteration t , the position can be updated by adding to a velocity v_i^t . Thus the male mayfly’s position update formula is:

$$x_i^{t+1} = x_i^t + v_i^t. \tag{1}$$

Consider those male mayflies are always performing nuptial dances not far from the water. The male mayfly’s velocity update formula is as follows:

$$v_{ij}^{t+1} = gv_{ij}^t + a_1 e^{-\beta r_p^2} (pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2} (gbest_j - x_{ij}^t), \tag{2}$$

where β is the visibility coefficient, a_1 and a_2 are defined as the positive constants representing the attraction. $pbest_{ij}$ is the best position obtained by the i th male mayfly in dimension j . r_p and r_g are the Euclidean distances between x_i and $pbest_i$ and between x_i and $gbest$, respectively. The gravity coefficient is given by g , which can be a fixed number between 0 and 1, or it can be steadily lowered over iterations as in Equation (3).

$$g = g_{\max} - \frac{g_{\max} - g_{\min}}{iter_{\max}} \times iter, \tag{3}$$

where g_{\max} and g_{\min} respectively represent the smallest and largest weights. The present and the total number of iterations are given by $iter$ and $iter_{\max}$, respectively.

The personal best position $pbest_i$ at iteration $t + 1$ is determined by Equation (4).

$$pbest_i = \begin{cases} x_i^{t+1}, & \text{if } f(x_i^{t+1}) < f(pbest_i) \\ \text{same as before,} & \text{otherwise.} \end{cases} \tag{4}$$

The best male mayflies keep performing up and down motions through different velocities. The velocities of these male mayflies are determined by Equation (5).

$$v_{ij}^{t+1} = v_{ij}^t + d \times r, \tag{5}$$

where the nuptial dance coefficient is given by d , and r means a random value in the range -1 to 1 .

2.1.2. Female Mayfly Flight

Female mayflies do not cluster together, however, they move towards male mayflies. y_i^t is defined as the present position of the i th female mayfly at iteration t . The change of the i th female mayfly in position is calculated as:

$$y_i^{t+1} = y_i^t + v_i^t. \tag{6}$$

In the MA, male and female mayflies with the same individual fitness ranking will attract each other, and the female mayfly’s position changes in response to the location of a male mayfly with the same ranking. The female mayfly’s velocity update formula is:

$$v_{ij}^{t+1} = \begin{cases} gv_{ij}^t + a_2 e^{-\beta r_{mf}^2} (x_{ij}^t - y_{ij}^t), & \text{if } f(y_i) > f(x_i) \\ gv_{ij}^t + fl \times r, & \text{if } f(y_i) \leq f(x_i), \end{cases} \quad (7)$$

where v_{ij}^t and y_{ij}^t are respectively defined as the i th female mayfly's velocity and position in dimension j at iteration t . a_2 and β are defined as the attraction constant and the visibility coefficient, respectively. r_{mf} is the Euclidean distance between the i th female mayfly and the i th male mayfly, fl is a random walk coefficient which suggests that a female is not attracted to a male, and r means a random value in the range -1 to 1 .

2.1.3. Mating Procedure

The mating is represented by the crossover operator which provides the global search capability for the MA. In the MA, male and female mayflies with the same individual fitness ranking will mate with each other to produce offspring mayflies. The mating operation of each pair of male and female mayflies produces two offspring, and the formula for the crossover operator is:

$$\begin{cases} \gamma_1 = L \times male + (1 - L) \times female \\ \gamma_2 = L \times female + (1 - L) \times male, \end{cases} \quad (8)$$

where L means a random value in the range 0 to 1 . Moreover, *male* and *female* denote the parents. Initially, the offspring's velocity is set to zero.

2.1.4. Mutate the Genes of Offspring

The mutation operator provides partial local search capability for MA. The chosen offspring can be updated by adding a random value to the variable throughout this process, so the offspring are changed to

$$\gamma'_n = \gamma_n + \sigma N_n(0, 1), \quad (9)$$

where the standard deviation and the standard normal distribution are represented by σ and N_n , respectively.

2.1.5. Reduction of Nuptial Dance and Random Walk

Although these operators provide the MA with powerful local search capability, d and fl must be gradually reduced. Both values can be updated over the iterations t using Equations (10) and (11), respectively.

$$d_t = d_0 \delta^t \quad (10)$$

$$fl_t = fl_0 \delta^t, \quad (11)$$

where δ is a fixed number between 0 and 1 .

As previously stated, Algorithm 1 depicts the MA's pseudo-code.

Algorithm 1 Pseudocode of MA.

Input: male and female population sizes, N_1 and N_2 ; maximum iterations, $iter_{max}$; visibility coefficient, β ; learning factors, a_1 and a_2 ; nuptial dance coefficient, d ; random flight coefficient, fl ; objective function, $f(x)$

Output: Optimal solution $gbest$

- 1: Initialize the male and female velocities v_m and v_f
- 2: Evaluate population based on $f(x)$
- 3: Find the global best ($gbest$)
- 4: **for** $iter = 1$ to $iter_{max}$ **do**
- 5: Adjust the speed and position of each female mayfly using Equations (6) and (7)
- 6: Adjust the speed and position of each male mayfly using Equations (1), (2) and (5)
- 7: Sort the mayflies and rank them based on $f(x)$
- 8: Perform crossover and generate male and female offspring
- 9: Mutate the offspring
- 10: Divide offspring into male and female at random
- 11: Update the worst old individuals with the finest new ones
- 12: Update $pbest$ and $gbest$
- 13: **end for**

2.2. UAV Path Planning Problem

UAV path planning aims to discover the best flight route for the UAV given a variety of complicated flying restrictions such as threat constraints and fuel limits [35]. The flying height of the drone might be constant along its flight path from the start point to the target point, so this paper only focuses on the UAV's horizontal flying distance. The mathematical model for UAV path planning and the cost function used to evaluate the planned path are described as follows.

2.2.1. Mathematical Model for Uav Path Planning

In this model, we define S and T to be the start point and target point of the UAV, respectively. It is crucial to prevent collisions with numerous obstacles during the flight of the drone. As a result, this paper employs circles of varying radii to substitute barriers of varying threat levels. So the task of a UAV flight is to find a flight path from S to T with the smallest overall cost while avoiding collisions with obstacles. Figure 1 depicts a schematic representation of the environment modeling for the UAV's two-dimensional path planning. The path connected by the red line segments is the feasible path.

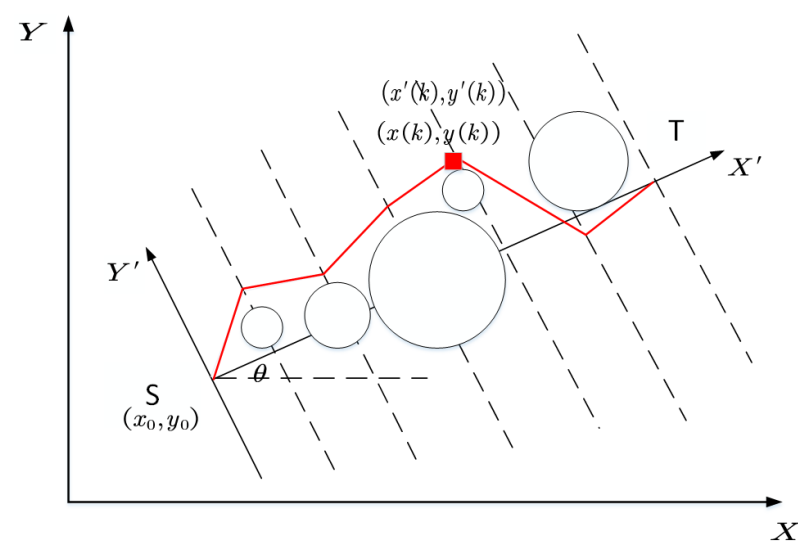


Figure 1. Two-dimensional representation of planning space.

The segment ST between the S and T in the original coordinate system O_{XY} is split into $D + 1$ equal portions by D vertical lines $\{L_k, k = 1, 2, \dots, D\}$. We can get the path from S to T by picking a point on each vertical line. Thus, the planned path can be expressed as:

$$L = \{S, (x(1), y(1)), \dots, (x(D), y(D)), T\}. \tag{12}$$

To accelerate the processing speed, let the segment ST be the horizontal axis and use Equation (13) to transfer each point $(x(k), y(k))$ in the O_{XY} to $(x'(k), y'(k))$ in the new coordinate system $O_{X'Y'}$.

$$\begin{bmatrix} x'(k) \\ y'(k) \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x(k) - x_s \\ y(k) - y_s \end{bmatrix} \tag{13}$$

where θ denotes to the rotation angle of the O_{XY} , and the point (x_s, y_s) is the coordinate in the O_{XY} .

As shown in Figure 1, the abscissa of each node in the $O_{X'Y'}$ can be calculated using formula $x'(k) = \frac{|ST|}{D+1} \times k$, so the planned path can be simplified and expressed as:

$$L' = \{(0, 0), (x'(1), y'(1)), \dots, (x'(D), y'(D)), (|ST|, 0)\}. \tag{14}$$

2.2.2. Fitness Function Modeling

A fitness function can be used to calculate the cost of a planned path. The fitness function value can be used to judge the quality of the planned path. The fitness function established in this paper is defined as the sum of two cost functions, given as follows:

$$F(L') = w_1 F_1(L') + w_2 F_2(L'), \tag{15}$$

where $F(L')$ denotes the total cost function of the planned path; $F_1(L')$ denotes the cost of the planned path's length; $F_2(L')$ denotes the smoothness cost of the planned path. w_1 and w_2 are the weight coefficients of the above cost function, and they are defined as:

$$\begin{cases} \omega_i \geq 0 \\ \sum_{i=1}^2 \omega_i = 1. \end{cases} \tag{16}$$

Given the UAV's restricted energy supply, the shorter the planned path length, the less time and energy the UAV spends, and the better it is for the UAV. Assume that the topography of the flight environment and information on the threatened area are known. The start state and the target state are L'_0 and L'_{D+1} , respectively. Hence, the cost F_1 related to the path length can be computed as:

$$F_1(L') = \sum_{k=0}^D d(L'_k, L'_{k+1}), \tag{17}$$

where $d(L'_k, L'_{k+1})$ is the distance between $(x'(k), y'(k))$ and $(x'(k + 1), y'(k + 1))$. If the waypoint falls into the obstacle, we choose the intersection of the vertical line with the obstacle, the point closest to the waypoint, as the new search agent. Figure 2a,b depicts the post-collision processing flow. The path connected by the red line segments is the modified feasible path.

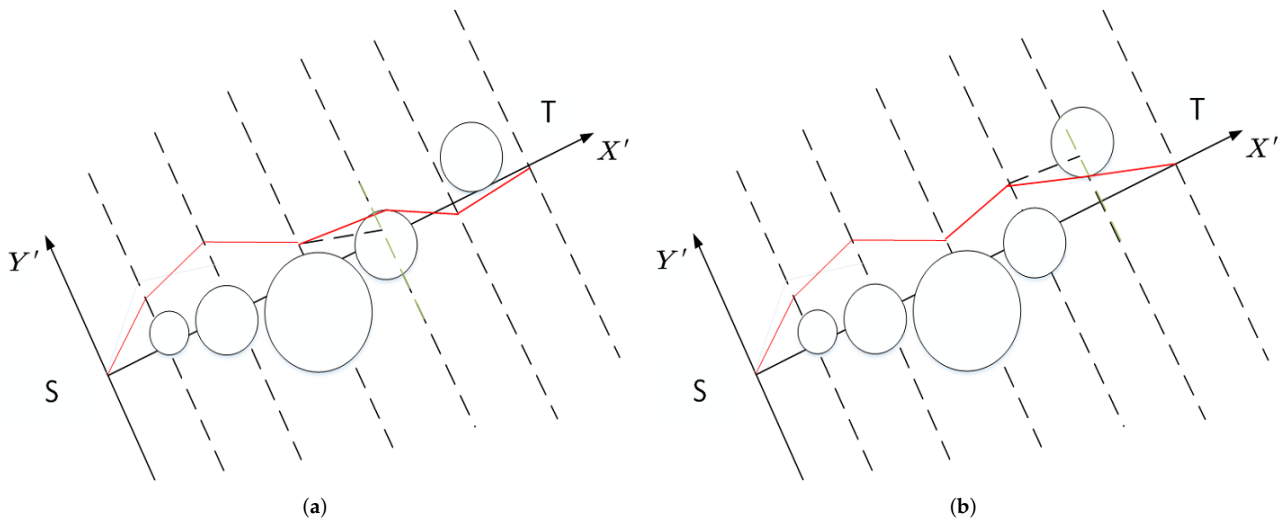


Figure 2. The post-collision processing flow. (a) Scenario 1; (b) Scenario 2.

The smoothness cost evaluates the turning rate critical to generating feasible paths. The smooth cost F_2 can be computed as:

$$\begin{cases} F_2(L') = \sum_{k=1}^D \Delta_k \\ \Delta_k = \begin{cases} \cos \varphi - \cos \theta_k, & \varphi \geq \theta_i \\ \varphi, & \varphi < \theta_i \end{cases} \\ \cos \theta = \frac{a_i^T a_{i+1}}{|a_i| |a_{i+1}|} \end{cases}, \quad (18)$$

where the φ is the maximum turning angle, θ is the current turning angle, and a_i is a vector representing the i th segment of the entire route.

3. Modified Mayfly Algorithm

The standard MA has a better convergence speed than other swarm algorithms when dealing with low-dimensional situations. However, due to the influence of velocity fluctuation, the stability of the MA is poor, thus it leads to poor results. Furthermore, when dealing with high-dimensional nonlinear complicated situations, the MA cannot merely rely on its mechanism to get out of the local optimal zone, so the MA performs poorly on the multimodal functions. This section proposes a modified mayfly algorithm named modMA, which combines three strategies to improve the MA. One improvement is to use the exponent decreasing inertia weight strategy to improve MA, another uses the adaptive Cauchy mutation strategy to improve MA, and a third uses an enhanced crossover operator to improve MA.

3.1. Improvement 1: Exponent Decreasing Inertia Weight Strategy

There are many decreasing inertia weight strategies, such as linear [36], adaptive [37], logarithmic [38], and others [39]. The larger inertia weight in the earlier period is favorable to the exploration of particles, and the particles can search a larger space. The smaller inertia weight in the later period is beneficial to the exploitation of particles. This paper introduces the exponent decreasing inertia weight [40] into MA, formula is as follows:

$$g = g_{\min} + \exp\left(1 - \frac{iter_{\max}}{iter_{\max} - iter + 1}\right) * (g_{\max} - g_{\min}), \quad (19)$$

where g_{\max} , g_{\min} , $iter$, and $iter_{\max}$ are the same to Equation (3).

3.2. Improvement 2: Adaptive Cauchy Mutation Strategy

When utilizing Equations (1) and (2) in the standard MA to change the velocity of each male mayfly, the diversity of the entire mayfly population may decline. To mitigate this problem, the position of each male mayfly should be readjusted by using Cauchy-based mutation [41,42]. The standard Cauchy distribution's probability density function can be described as:

$$f(x) = \frac{1}{\pi} \times \frac{1}{1+x^2}, \quad (20)$$

where $-\infty < x < \infty$, the corresponding distribution function is expressed as:

$$F(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}. \quad (21)$$

Random numbers generated via the standard Cauchy distribution are shown below:

$$CM = \tan\left(\pi \times \left(\varepsilon - \frac{1}{2}\right)\right), \quad (22)$$

where CM denotes a random value produced by the inverse function of the standard Cauchy distribution function, ε means a random value in the range 0 to 1 produced by a uniform distribution.

According to Fogarty [43], improved performance is attained when the mutation rate falls exponentially with the number of iterations. Ref. [44] pointed out that the concept is related to that of the simulated annealing (SA) algorithm [45]. So after utilizing Equations (1) and (2) to change the position of each male mayfly, the position of each male mayfly needs to be readjusted using the Equation (22), which combines Cauchy mutation strategy with an adaptive mutation.

$$X_i^{t+1} = x_i^{t+1} + x_i^{t+1} \times CM \times \exp((1-t) \times \alpha), \quad (23)$$

where x_i^{t+1} is the i th individual calculated using Equations (1) and (2), and X_i^{t+1} represents the mutated individual. α is a contraction-expansion coefficient that controls the speed of convergence and it is defined as 0.15.

In this section, we combine the Cauchy mutation strategy with an adaptive mutation which can make the MA generate perturbations with a large step size in the early iterations so that the particles can jump from their current position to another and help MA escape the local optima. The perturbations with a smaller step size in the late iterations can speed up the convergence of the MA. It is worth mentioning that we do not use Cauchy-based mutation to readjust the positions of the particles calculated using Equations (1) and (5).

3.3. Improvement 3: Enhanced Crossover Operator

Although the crossover and reproduction operations of mayflies provide the MA with exploration ability as well as enhance the MA's convergence speed, the MA has a not strong exploration ability. Hence, we use the horizontal crossover search [46] to improve it. The enhanced crossover operator of the MA employing horizontal crossover search is as follows:

$$\begin{cases} \gamma_1 = L \times male + (1-L) \times female + c_1 \times (male - female) \\ \gamma_2 = L \times female + (1-L) \times male + c_2 \times (female - male), \end{cases} \quad (24)$$

where c_1 and c_2 are expansion coefficients, and they are both random values between -1 and 1 produced by a uniform distribution.

The horizontal crossover operation divides the problem-solving space of multi-dimensional into half-population of hypercubes, allowing new spots on the hypercube's periphery to be sampled with a low probability, and reducing the blind region that are unable to explore by two paired parent individuals. Offspring generated using the horizontal

crossover operation need to be compared with their parents, and individuals with greater fitness values are selected for the following iteration. It causes the MA to continuously converge to the best solution and ensure the convergence efficiency without affecting the optimization accuracy through the horizontal crossover operation.

We can also appropriately shrink or expand the search space generated by the two paired parent individuals. The formula of the crossover operator of the MA improved by shrinking the search space is as follows:

$$\begin{cases} \gamma_1 = d_1 \times (L \times \text{male} + (1 - L) \times \text{female}) \\ \gamma_2 = d_3 \times (L \times \text{female} + (1 - L) \times \text{male}), \end{cases} \quad (25)$$

where d_1 and d_3 are the contraction coefficients, and they are both random values between 0.7 and 1 produced by a uniform distribution.

The formula of the crossover operator of the MA improved by expanding the search space is as follows:

$$\begin{cases} \gamma_1 = d_2 \times (L \times \text{male} + (1 - L) \times \text{female}) \\ \gamma_2 = d_4 \times (L \times \text{female} + (1 - L) \times \text{male}), \end{cases} \quad (26)$$

where d_2 and d_4 are the expansion coefficients, and they are both random values between 1 and 1.3 produced by a uniform distribution.

Therefore, the enhanced crossover operator formula of MA can be described as:

$$\gamma_1 = \begin{cases} L \times \text{male} + (1 - L) \times \text{female}, \text{rand}_1 < p_{\text{one}} \\ L \times \text{male} + (1 - L) \times \text{female} + c_1 \times (\text{male} - \text{female}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}} \\ d_1 \times (L \times \text{male} + (1 - L) \times \text{female}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}}, \text{rand}_3 < p_{\text{three}} \\ d_3 \times (L \times \text{male} + (1 - L) \times \text{female}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}}, \text{rand}_3 > p_{\text{three}}. \end{cases} \quad (27)$$

$$\gamma_2 = \begin{cases} L \times \text{female} + (1 - L) \times \text{male}, \text{rand}_1 < p_{\text{one}} \\ L \times \text{female} + (1 - L) \times \text{male} + c_2 \times (\text{female} - \text{male}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}} \\ d_2 \times (L \times \text{female} + (1 - L) \times \text{male}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}}, \text{rand}_3 < p_{\text{three}} \\ d_4 \times (L \times \text{female} + (1 - L) \times \text{male}), \text{rand}_1 > p_{\text{one}}, \text{rand}_2 < p_{\text{two}}, \text{rand}_3 > p_{\text{three}}. \end{cases} \quad (28)$$

where rand_1 , rand_2 , and rand_3 are the random input numbers between [0,1]. p_{one} , p_{two} , and p_{three} are the switching probabilities, which control the method of offspring generation. For the value of switching probability p_{one} , we refer to the parameter tuning technique of switching probability in the Butterfly Optimization Algorithm (BOA) [47]. From our simulations, we found that $p_{\text{one}} = 0.8$, $p_{\text{two}} = 0.5$, and $p_{\text{three}} = 0.5$ work better for most applications.

4. Experimental Results and Application

In this part, we select a variety of numerical optimization functions from CEC benchmark functions to assess the effect of the modified MA presented in this article. Tables A1 and A2 in Appendix A show the expression, dimension, domain of a variable, and minimum value of each benchmark function. These benchmark functions used in this study can be found in [48], and they are classified into two groups. The unimodal benchmark functions (F1–F14) and multimodal benchmark functions (F15–F26) are applied to calculate the abilities to exploit and explore of the algorithm benchmarked by benchmark functions, respectively.

To assess the efficacy of the three improvement techniques described in this study for MA, we first test and analyze the three versions of the improved MA through comparison experiment 1 by six benchmark functions from Tables A1 and A2 with $\text{Dim} = 100$ and $\text{Dim} = 300$. It is worth mentioning that modMA-1 is characterized in this study as an MA that is improved by an adaptive Cauchy mutation strategy and the original crossover operator, modMA-2 is defined as an MA modified using only the enhanced crossover operator, and modMA is defined as an MA that incorporates three improved strategies.

To demonstrate that the proposed modMA outperforms other algorithms, we design the comparison experiment 2 for MA [24], modMA-1, modMA-2, modMA, PSO [11], GWO [20], and BOA [47] for 26 benchmark functions with $Dim = 50$. Finally, we apply the three versions of the modified MA to the UAV path planning problem.

4.1. Parameter Settings of Seven Algorithms in Test

In the two experiments, we select MA [24], modMA-1, modMA-2, modMA, PSO [11], GWO [20], and BOA [47] for comparison. Table 1 shows the parameter settings of each algorithm. To promote equality in the comparison, all algorithms in this study are iterated 1000 times, the dimension is 50, and each algorithm has a maximum number of solutions of 40. For example, we use 20 female mayflies and 20 male mayflies for all simulations of MA. We analyze the Average (Avg) and Standard deviation (Std) of the fitness values after 30 runs on each benchmark function in both experiments.

4.2. Experiment 1: Comparison with MA, modMA-1, modMA-2,

In experiment 1, we compare the performance of MA [24], modMA-1, modMA-2, and modMA on six high-dimensional functions (3 unimodal problems and 3 multimodal problems) from Tables A1 and A2 with $Dim = 100$ and $Dim = 300$. It can be seen from Figures 3 and 4a–f that the proposed modMA, modMA-1, modMA-2 have improved the convergence trend of the MA at high dimensions (i.e., 300 and 500), that is, modMA, modMA-1, modMA-2 can obtain better values on the benchmark functions with fewer iterations. From Figure 3c,e,f we can see that modMA has a certain level of ability in escaping from local optima values in the late iterations. From Tables 2 and 3, the proposed modMA, modMA-1, modMA-2 can also get better results. From the Avg and Std in Tables 2 and 3, the modMA can solve the problems that MA has poor ability and instability in finding the optimal solution when solving high-dimensional complex functions.

Table 1. Parameter settings for the algorithms to be compared.

| No. | Name | Parameter |
|-----|---------|---|
| 1 | MA | $g_{max} = 0.9, g_{min} = 0.2, a_1 = 1, a_2 = a_3 = 1.5, d = 5, \beta = 2, fl = 1$ |
| 2 | modMA-1 | $g_{max} = 0.9, g_{min} = 0.2, a_1 = 1, a_2 = a_3 = 1.5, d = 5, \beta = 2, fl = 1, \alpha = 0.15$ |
| 3 | modMA-2 | $g_{max} = 0.9, g_{min} = 0.2, a_1 = 1, a_2 = a_3 = 1.5, d = 5, \beta = 2, fl = 1$ |
| 4 | modMA | $g_{max} = 0.9, g_{min} = 0.2, a_1 = 1, a_2 = a_3 = 1.5, d = 5, \beta = 2, fl = 1, \alpha = 0.15$ |
| 5 | PSO | $g_{max} = 0.9, g_{min} = 0.2, c_1 = c_2 = 1.5$ |
| 6 | GWO | $a_{first} = 2, a_{final} = 0$ |
| 7 | BOA | $probabilityswitch = 0.8, powerexponent = 0.1, modularmodality = 0.01$ |

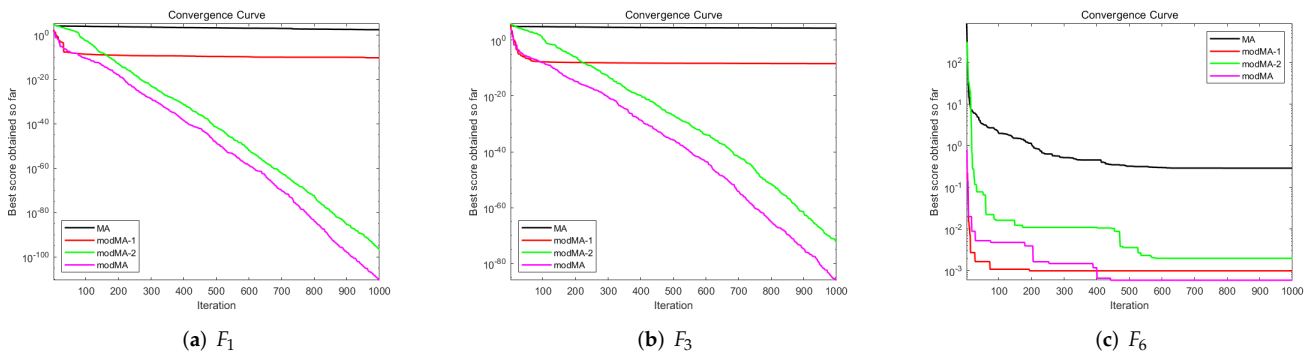


Figure 3. Cont.

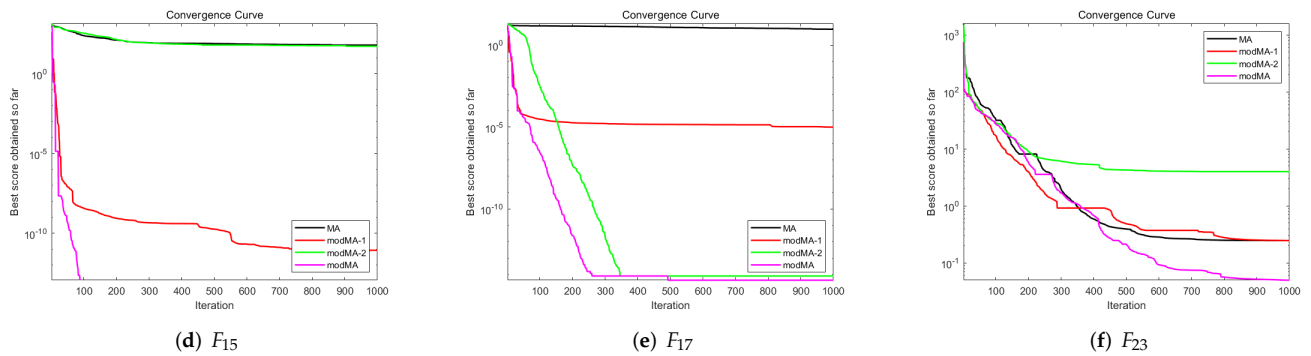


Figure 3. Comparison of convergence curve between MA, modMA-1, modMA-2, and modMA with $Dim = 100$.

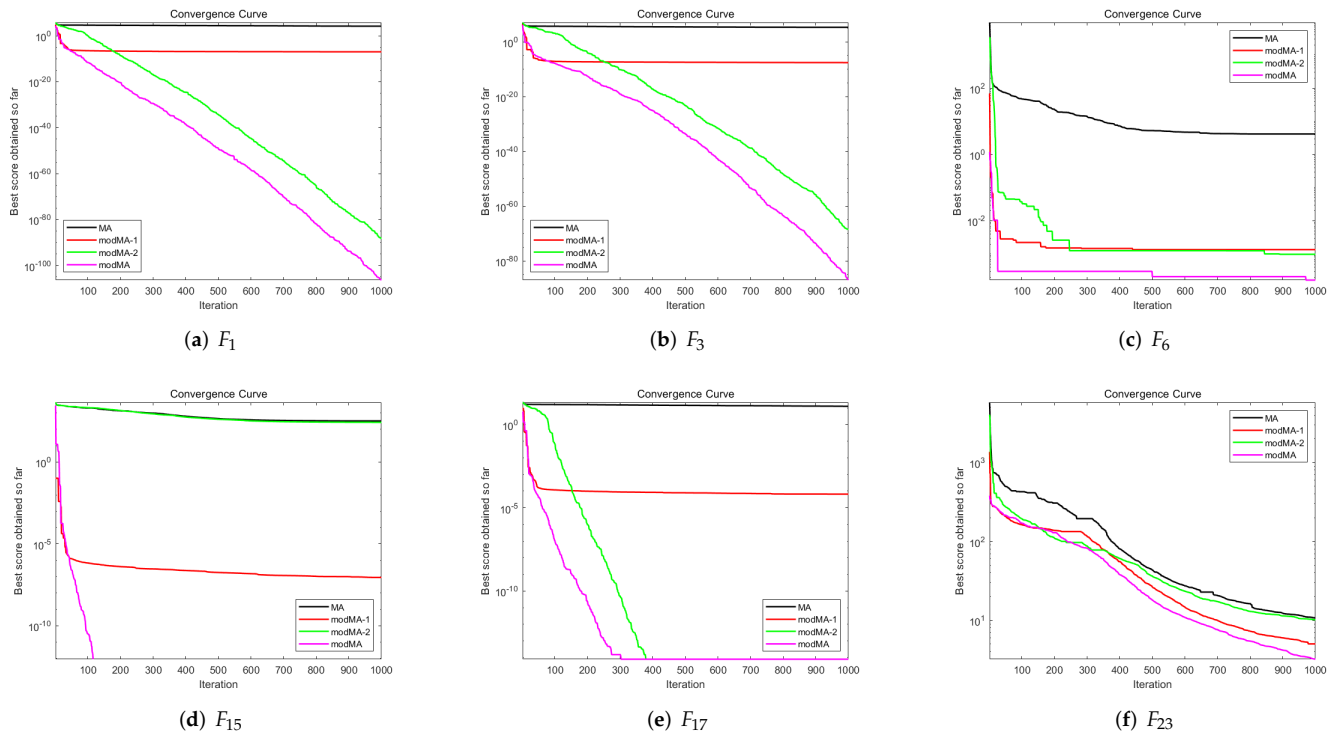


Figure 4. Comparison of convergence curve between MA, modMA-1, modMA-2, and modMA with $Dim = 300$.

4.3. Experiment 2: Comparison with Other Swarm Algorithms

The performance of MA [24], modMA-1, modMA-2, modMA, PSO [11], GWO [20], and BOA [47] for 26 benchmark functions with $Dim = 50$ is compared in this experiment. At the same time, the *Avg* and the *Std* of functions are shown in Table 4.

From Table 4, the proposed modMA, modMA-1, and modMA-2 all perform well on most benchmark functions when *Avg* is used as a comparison criterion.

modMA has the best performance on these 26 benchmarks functions with $Dim = 50$ except $F_4, F_5, F_7, F_{20}, F_{21}$ and F_{23} . modMA can obtain the minimum values on benchmark functions $F_{13}, F_{14}, F_{15}, F_{16}, F_{18}, F_{24}, F_{26}$.

The Friedman test values and Wilcoxon sign rank test values that can reflect the difference between the proposed modMA and other algorithms (i.e., MA, modMA-1, modMA-2, PSO, GWO, and BOA) are also given in Tables 5 and 6. Note, the significant level α is set to 0.05. This signifies that the sample result is not statistically significant if the p-value is larger than 0.05.

In addition, we also select the convergence curves of various algorithms on some benchmark functions to visually show the degree of convergence of the modMA-1, modMA-2, and modMA, as shown in Figures 5 and 6. Compared with MA, the convergence accuracy of the proposed modMA-1, modMA-2, and modMA is significantly improved. Compared to modMA-1 and modMA-2, modMA converges faster and finds the global optimum with fewer iterations in most functions. On the whole, modMA can effectively improve the capability of exploitation, exploration, and stability of MA. modMA also has a better convergence rate and accuracy than other algorithms.

Table 2. MA, modMA-1, modMA-2, and modMA test results on the 100-dimensional functions in experiment 1.

| Functions | | MA | modMA-1 | modMA-2 | modMA |
|-----------------|------|---------------------------|---------------------------|---------------------------------|----------------------------------|
| F ₁ | Best | 1.130 × 10 ² | 4.798 × 10 ⁻¹⁶ | 6.615 × 10 ⁻¹⁰⁰ | 1.443 × 10 ⁻¹¹³ |
| | Avg | 2.586 × 10 ² | 5.163 × 10 ⁻⁹ | 2.228 × 10 ⁻⁹⁰ | 5.000 × 10⁻¹⁰² |
| | Std | 1.033 × 10 ² | 1.320 × 10 ⁻⁸ | 5.748 × 10 ⁻⁹⁰ | 2.713 × 10 ⁻¹⁰¹ |
| F ₃ | Best | 1.049 × 10 ⁴ | 2.301 × 10 ⁻¹⁴ | 3.723 × 10 ⁻⁸¹ | 3.236 × 10 ⁻⁹⁸ |
| | Avg | 2.029 × 10 ⁴ | 1.035 × 10 ⁻⁵ | 1.499 × 10 ⁻⁶⁷ | 1.783 × 10⁻⁸¹ |
| | Std | 5.771 × 10 ³ | 2.889 × 10 ⁻⁵ | 8.154 × 10 ⁻⁶⁷ | 9.097 × 10 ⁻⁸¹ |
| F ₆ | Best | 1.517 × 10 ⁻¹ | 1.586 × 10 ⁻⁴ | 3.283 × 10 ⁻⁴ | 6.919 × 10 ⁻⁵ |
| | Avg | 92.821 × 10 ⁻¹ | 1.920 × 10 ⁻³ | 1.120 × 10 ⁻³ | 8.239 × 10⁻⁴ |
| | Std | 6.322 × 10 ⁻² | 2.029 × 10 ⁻³ | 5.825 × 10 ⁻⁴ | 5.655 × 10 ⁻⁴ |
| F ₁₅ | Best | 4.179 × 10 ¹ | 2.274 × 10 ⁻¹³ | 3.084 × 10 ¹ | 0.000 × 10 ⁰ |
| | Avg | 5.948 × 10 ¹ | 5.983 × 10 ⁻¹¹ | 5.799 × 10 ¹ | 0.000 × 10⁰ |
| | Std | 9.749 × 10 ⁰ | 1.629 × 10 ⁻¹⁰ | 1.473 × 10 ¹ | 0.000 × 10 ⁰ |
| F ₁₇ | Best | 7.314 × 10 ⁰ | 8.795 × 10 ⁻¹⁰ | 4.441 × 10 ⁻¹⁵ | 4.441 × 10 ⁻¹⁵ |
| | Avg | 9.007 × 10 ⁰ | 1.006 × 10 ⁻⁵ | 5.862 × 10⁻¹⁵ | 5.862 × 10⁻¹⁵ |
| | Std | 9.048 × 10 ⁻¹ | 1.516 × 10 ⁻⁵ | 1.770 × 10 ⁻¹⁵ | 1.770 × 10 ⁻¹⁵ |
| F ₂₃ | Best | 1.762 × 10 ⁻² | 1.652 × 10 ⁻² | 1.483 × 10 ⁻² | 1.366 × 10 ⁻² |
| | Avg | 6.624 × 10 ⁻¹ | 4.767 × 10 ⁻¹ | 3.482 × 10 ⁻¹ | 2.002 × 10⁻¹ |
| | Std | 1.275 × 10 ⁰ | 6.518 × 10 ⁻¹ | 8.994 × 10 ⁻¹ | 3.296 × 10 ⁻¹ |

Table 3. MA, modMA-1, modMA-2, and modMA test results on the 300-dimensional functions in experiment 1.

| Functions | | MA | modMA-1 | modMA-2 | modMA |
|-----------------|------|--------------------------|---------------------------|---------------------------------|---------------------------------|
| F ₁ | Best | 1.223 × 10 ⁴ | 4.977 × 10 ⁻¹² | 4.135 × 10 ⁻⁹⁷ | 1.469 × 10 ⁻¹⁰⁹ |
| | Avg | 1.607 × 10 ⁴ | 1.547 × 10 ⁻⁶ | 9.198 × 10 ⁻⁸⁷ | 4.037 × 10⁻⁹² |
| | Std | 1.865 × 10 ³ | 5.898 × 10 ⁻⁶ | 3.814 × 10 ⁻⁸⁶ | 2.206 × 10 ⁻⁹¹ |
| F ₃ | Best | 1.405 × 10 ⁵ | 1.363 × 10 ⁻⁸ | 9.877 × 10 ⁻⁷⁷ | 1.422 × 10 ⁻⁹¹ |
| | Avg | 2.414 × 10 ⁵ | 1.168 × 10 ⁻³ | 4.772 × 10 ⁻⁶³ | 2.737 × 10⁻⁷⁷ |
| | Std | 8.673 × 10 ⁴ | 4.095 × 10 ⁻³ | 2.340 × 10 ⁻⁶² | 1.495 × 10 ⁻⁷⁶ |
| F ₆ | Best | 2.985 × 10 ⁰ | 1.681 × 10 ⁻⁴ | 2.015 × 10 ⁻⁴ | 6.201 × 10 ⁻⁵ |
| | Avg | 3.827 × 10 ⁰ | 2.248 × 10 ⁻³ | 1.273 × 10 ⁻³ | 8.106 × 10⁻⁴ |
| | Std | 5.001 × 10 ⁻¹ | 2.093 × 10 ⁻³ | 6.273 × 10 ⁻⁴ | 5.993 × 10 ⁻⁴ |
| F ₁₅ | Best | 1.868 × 10 ² | 6.366 × 10 ⁻¹² | 1.249 × 10 ² | 0.000 × 10 ⁰ |
| | Avg | 2.351 × 10 ² | 2.824 × 10 ⁻⁷ | 2.269 × 10 ² | 0.000 × 10⁰ |
| | Std | 2.567 × 10 ¹ | 1.187 × 10 ⁻⁶ | 4.775 × 10 ¹ | 0.000 × 10 ⁰ |
| F ₁₇ | Best | 1.245 × 10 ¹ | 9.821 × 10 ⁻⁷ | 4.441 × 10 ⁻¹⁵ | 4.441 × 10 ⁻¹⁵ |
| | Avg | 1.337 × 10 ¹ | 7.345 × 10 ⁻⁵ | 6.099 × 10⁻¹⁵ | 6.099 × 10⁻¹⁵ |
| | Std | 4.853 × 10 ⁻¹ | 1.129 × 10 ⁻⁴ | 1.741 × 10 ⁻¹⁵ | 1.803 × 10 ⁻¹⁵ |
| F ₂₃ | Best | 4.761 × 10 ⁰ | 2.691 × 10 ⁰ | 3.108 × 10 ⁰ | 2.294 × 10 ⁰ |
| | Avg | 7.554 × 10 ⁰ | 4.673 × 10 ⁰ | 5.422 × 10 ⁰ | 3.979 × 10⁰ |
| | Std | 1.804 × 10 ⁰ | 1.061 × 10 ⁰ | 1.415 × 10 ⁰ | 8.642 × 10 ⁻¹ |

Table 4. MA, modMA-1, modMA-2, modMA, PSO, GWO, and BOA test results on the 50-dimensional functions in experiment 2.

| Functions | | MA | modMA-1 | modMA-2 | modMA | PSO | GWO | BOA |
|-----------------|------|--------------------------------|---------------------------------|---------------------------------|----------------------------------|--------------------------------|--------------------------------|---------------------------|
| F ₁ | Mean | 4.778 × 10 ⁻⁵ | 5.874 × 10 ⁻¹³ | 2.055 × 10 ⁻⁹⁰ | 4.660 × 10⁻¹⁰⁶ | 3.555 × 10 ⁻³ | 2.875 × 10 ⁻⁴⁸ | 1.761 × 10 ⁻¹⁴ |
| | Std | 1.533 × 10 ⁻⁴ | 1.598 × 10 ⁻¹² | 1.119 × 10 ⁻⁸⁹ | 1.82 × 10 ⁻¹⁰⁵ | 9.273 × 10 ⁻³ | 2.797 × 10 ⁻⁴⁸ | 7.850 × 10 ⁻¹⁶ |
| F ₂ | Mean | 4.695 × 10 ⁻³ | 7.326 × 10 ⁻⁹ | 4.029 × 10 ⁻⁵¹ | 4.294 × 10⁻⁵⁵ | 2.038 × 10 ⁻² | 1.406 × 10 ⁻²⁸ | 2.791 × 10 ²² |
| | Std | 7.433 × 10 ⁻³ | 1.632 × 10 ⁻⁸ | 7.881 × 10 ⁻⁵¹ | 1.834 × 10 ⁻⁵⁴ | 3.023 × 10 ⁻² | 1.342 × 10 ⁻²⁸ | 8.588 × 10 ²² |
| F ₃ | Mean | 1.978 × 10 ³ | 8.022 × 10 ⁻⁸ | 3.506 × 10 ⁻⁷⁰ | 2.191 × 10⁻⁸³ | 2.306 × 10 ³ | 2.120 × 10 ⁻⁸ | 1.773 × 10 ⁻¹⁴ |
| | Std | 8.416 × 10 ² | 2.235 × 10 ⁻⁷ | 1.920 × 10 ⁻⁶⁹ | 1.015 × 10 ⁻⁸² | 7.139 × 10 ² | 6.684 × 10 ⁻⁸ | 1.050 × 10 ⁻¹⁵ |
| F ₄ | Mean | 1.693 × 10 ⁰ | 6.127 × 10 ⁻⁶ | 2.147 × 10⁻⁴¹ | 3.468 × 10 ⁻⁴¹ | 6.888 × 10 ⁻¹ | 1.802 × 10 ⁻¹¹ | 1.057 × 10 ⁻¹¹ |
| | Std | 3.424 × 10 ⁻¹ | 1.037 × 10 ⁻⁵ | 9.710 × 10 ⁻⁴¹ | 1.900 × 10 ⁻⁴⁰ | 8.100 × 10 ⁻² | 2.251 × 10 ⁻¹¹ | 4.996 × 10 ⁻¹³ |
| F ₅ | Mean | 4.184 × 10⁻⁸ | 4.388 × 10 ⁻⁸ | 4.502 × 10 ⁻⁸ | 5.995 × 10 ⁻⁸ | 4.134 × 10 ⁻⁵ | 1.873 × 10 ⁰ | 1.031 × 10 ¹ |
| | Std | 1.871 × 10 ⁻⁸ | 2.737 × 10 ⁻⁸ | 3.135 × 10 ⁻⁸ | 2.724 × 10 ⁻⁸ | 1.382 × 10 ⁻⁴ | 5.635 × 10 ⁻¹ | 7.762 × 10 ⁻¹ |
| F ₆ | Mean | 6.249 × 10 ⁻² | 1.826 × 10 ⁻³ | 1.145 × 10 ⁻³ | 6.490 × 10⁻⁴ | 3.759 × 10 ⁻² | 1.010 × 10 ⁻³ | 6.995 × 10 ⁻⁴ |
| | Std | 1.634 × 10 ⁻² | 1.533 × 10 ⁻³ | 6.072 × 10 ⁻⁴ | 5.269 × 10 ⁻⁴ | 1.022 × 10 ⁻² | 3.972 × 10 ⁻⁴ | 2.370 × 10 ⁻⁴ |
| F ₇ | Mean | 2.67 × 10 ⁻¹⁰⁹ | 2.67 × 10 ⁻¹⁰⁹ | 2.67 × 10 ⁻¹⁰⁹ | 2.67 × 10 ⁻¹⁰⁹ | 0.000 × 10⁰ | 5.688 × 10 ⁻⁹¹ | 4.816 × 10 ⁻¹⁴ |
| | Std | 9.62 × 10 ⁻¹²⁵ | 9.62 × 10 ⁻¹²⁵ | 9.62 × 10 ⁻¹²⁵ | 9.62 × 10 ⁻¹²⁵ | 0.000 × 10 ⁰ | 2.500 × 10 ⁻⁹⁰ | 2.567 × 10 ⁻¹³ |
| F ₈ | Mean | 9.117 × 10 ⁻²⁴ | 8.800 × 10 ⁻⁴⁷ | 5.99 × 10 ⁻²⁵⁰ | 1.345 × 10⁻²⁶⁵ | 1.431 × 10 ⁻²⁷ | 2.35 × 10 ⁻²⁰² | 7.036 × 10 ⁻¹⁷ |
| | Std | 3.569 × 10 ⁻²³ | 4.504 × 10 ⁻⁴⁶ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 6.681 × 10 ⁻²⁷ | 0.000 × 10 ⁰ | 8.769 × 10 ⁻¹⁷ |
| F ₉ | Mean | 1.857 × 10 ⁻⁶ | 8.590 × 10 ⁻¹⁵ | 9.788 × 10 ⁻⁹⁵ | 8.851 × 10⁻¹⁰⁷ | 5.935 × 10 ⁻⁴ | 5.060 × 10 ⁻⁴⁹ | 1.779 × 10 ⁻¹⁴ |
| | Std | 2.261 × 10 ⁻⁶ | 2.701 × 10 ⁻¹⁴ | 5.349 × 10 ⁻⁹⁴ | 4.84 × 10 ⁻¹⁰⁶ | 7.443 × 10 ⁻⁴ | 9.881 × 10 ⁻⁴⁹ | 9.194 × 10 ⁻¹⁶ |
| F ₁₀ | Mean | 4.387 × 10 ¹ | 4.432 × 10 ¹ | 4.232 × 10 ¹ | 4.214 × 10¹ | 8.405 × 10 ¹ | 4.691 × 10 ¹ | 4.865 × 10 ¹ |
| | Std | 4.150E × 10 ¹ | 5.236 × 10 ⁻¹ | 5.382 × 10 ⁻¹ | 4.549 × 10 ⁻¹ | 5.093 × 10 ¹ | 8.848 × 10 ⁻¹ | 3.134 × 10 ⁻² |
| F ₁₁ | Mean | 1.188 × 10 ⁻⁶ | 3.801 × 10 ⁻¹² | 5.403 × 10 ⁻⁹⁷ | 1.677 × 10⁻¹⁰⁸ | 3.334 × 10 ⁻⁴ | 3.162 × 10 ⁻⁴⁸ | 1.781 × 10 ⁻¹⁴ |
| | Std | 5.051 × 10 ⁻⁷ | 2.081 × 10 ⁻¹¹ | 2.684 × 10 ⁻⁹⁶ | 6.50 × 10 ⁻¹⁰⁸ | 3.967 × 10 ⁻⁴ | 5.144 × 10 ⁻⁴⁸ | 7.002 × 10 ⁻¹⁶ |
| F ₁₂ | Mean | 3.576 × 10 ⁰ | 6.667 × 10⁻¹ | 6.667 × 10⁻¹ | 6.667 × 10⁻¹ | 4.652 × 10 ⁰ | 6.667 × 10⁻¹ | 9.938 × 10 ⁻¹ |
| | Std | 3.066 × 10 ⁰ | 1.938 × 10 ⁻⁷ | 1.899 × 10 ⁻⁷ | 3.630 × 10 ⁻⁷ | 4.149 × 10 ⁰ | 5.015 × 10 ⁻⁷ | 2.995 × 10 ⁻³ |
| F ₁₃ | Mean | 0.000 × 10⁰ | 0.000 × 10⁰ | 0.000 × 10⁰ | 0.000 × 10⁰ | 2.10 × 10 ⁻¹⁴⁷ | 0.000 × 10⁰ | 8.645 × 10 ⁻²³ |
| | Std | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 1.13 × 10 ⁻¹⁴⁶ | 0.000 × 10 ⁰ | 3.426 × 10 ⁻²² |
| F ₁₄ | Mean | 2.423 × 10 ⁻² | 0.000 × 10⁰ | 0.000 × 10⁰ | 0.000 × 10⁰ | 2.96 × 10 ⁻¹¹⁹ | 0.000 × 10⁰ | 5.324 × 10 ⁻¹⁵ |
| | Std | 1.327 × 10 ⁻¹ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 1.62 × 10 ⁻¹¹⁸ | 0.000 × 10 ⁰ | 5.818 × 10 ⁻¹⁵ |
| F ₁₅ | Mean | 2.164 × 10 ¹ | 2.634 × 10 ⁻¹³ | 2.189 × 10 ¹ | 0.000 × 10⁰ | 9.333 × 10 ¹ | 2.776 × 10 ⁻¹ | 3.790 × 10 ⁻¹⁵ |
| | Std | 4.315 × 10 ⁰ | 1.725 × 10 ⁻¹³ | 5.793 × 10 ⁻⁰ | 0.000 × 10 ⁰ | 2.150 × 10 ¹ | 1.073 × 10 ⁰ | 2.076 × 10 ⁻¹⁴ |
| F ₁₆ | Mean | 2.080 × 10 ¹ | 3.878 × 10 ⁻¹⁴ | 2.167 × 10 ¹ | 0.000 × 10⁰ | 9.597 × 10 ¹ | 2.067 × 10 ⁰ | 4.049 × 10 ¹ |
| | Std | 8.058 × 10 ⁰ | 1.339 × 10 ⁻¹³ | 7.864 × 10 ⁰ | 0.000 × 10 ⁰ | 2.736 × 10 ¹ | 3.886 × 10 ⁰ | 1.051 × 10 ² |
| F ₁₇ | Mean | 5.000 × 10 ⁰ | 1.751 × 10 ⁻⁷ | 6.099 × 10 ⁻¹⁵ | 5.507 × 10⁻¹⁵ | 8.183 × 10 ⁻¹ | 3.490 × 10 ⁰ | 1.158 × 10 ⁻¹² |
| | Std | 8.102 × 10 ⁻¹ | 8.547 × 10 ⁻⁷ | 1.803 × 10 ⁻¹⁵ | 1.656 × 10 ⁻¹⁵ | 6.928 × 10 ⁻¹ | 7.937 × 10 ⁰ | 1.646 × 10 ⁻¹² |
| F ₁₈ | Mean | 1.386 × 10 ⁰ | 1.184 × 10 ⁻¹³ | 9.036 × 10 ⁻⁴ | 0.000 × 10⁰ | 1.534 × 10 ⁻² | 0.000 × 10⁰ | 3.938 × 10 ⁻¹⁵ |
| | Std | 2.809 × 10 ⁻¹ | 5.123 × 10 ⁻¹³ | 2.856 × 10 ³ | 0.000 × 10 ⁰ | 2.036 × 10 ⁻² | 0.000 × 10 ⁰ | 3.047 × 10 ⁻¹⁵ |
| F ₁₉ | Mean | 2.890 × 10 ⁻³ | 1.782 × 10 ⁻⁹ | 2.272 × 10 ⁻⁴ | 3.265 × 10⁻⁵⁷ | 5.072 × 10 ⁻³ | 1.060 × 10 ⁻⁴ | 1.652 × 10 ⁻¹⁴ |
| | Std | 1.243 × 10 ⁻² | 3.270 × 10 ⁻⁹ | 7.865 × 10 ⁻⁴ | 1.437 × 10 ⁻⁵⁶ | 4.089 × 10 ⁻³ | 2.878 × 10 ⁻⁴ | 1.204 × 10 ⁻¹⁴ |
| F ₂₀ | Mean | 1.473 × 10 ⁰ | 3.536 × 10⁻¹⁰ | 1.040 × 10 ⁻² | 5.339 × 10 ⁻¹⁰ | 5.595 × 10 ⁻¹ | 7.583 × 10 ⁻² | 8.704 × 10 ⁻¹ |
| | Std | 8.628 × 10 ⁻¹ | 2.116 × 10 ⁻¹⁰ | 3.692 × 10 ⁻² | 2.591 × 10 ⁻¹⁰ | 5.458 × 10 ⁻¹ | 2.722 × 10 ⁻² | 1.443 × 10 ⁻¹ |
| F ₂₁ | Mean | 1.095 × 10 ¹ | 4.040 × 10 ⁻¹ | 1.986 × 10⁻¹ | 6.401 × 10 ⁻¹ | 2.134 × 10 ⁰ | 1.435 × 10 ⁰ | 4.954 × 10 ⁰ |
| | Std | 7.068 × 10 ⁰ | 3.868 × 10 ⁻¹ | 3.389 × 10 ⁻¹ | 5.245 × 10 ⁻¹ | 2.113 × 10 ⁰ | 3.360 × 10 ⁻¹ | 1.596 × 10 ⁻¹ |
| F ₂₂ | Mean | 6.665 × 10 ⁻² | 2.451 × 10 ⁻¹⁰ | 2.741 × 10 ⁻³ | 1.077 × 10⁻⁸⁰ | 1.316 × 10 ⁻¹ | 1.513 × 10 ⁻³⁶ | 1.302 × 10 ⁻²¹ |
| | Std | 2.995 × 10 ⁻¹ | 8.827 × 10 ⁻¹⁰ | 5.282 × 10 ⁻³ | 3.542 × 10 ⁻⁸⁰ | 1.770 × 10 ⁻¹ | 3.101 × 10 ⁻³⁶ | 3.949 × 10 ⁻²¹ |
| F ₂₃ | Mean | 1.836 × 10 ⁻¹ | 1.769 × 10 ⁻¹ | 1.900 × 10 ⁻¹ | 1.395 × 10 ⁻¹ | 8.792 × 10⁻² | 6.202 × 10 ⁰ | 2.488 × 10 ¹ |
| | Std | 5.037 × 10 ⁻¹ | 3.862 × 10 ⁻¹ | 3.908 × 10 ⁻¹ | 1.084 × 10 ⁻¹ | 2.203 × 10 ⁻¹ | 2.122 × 10 ⁰ | 3.279 × 10 ⁰ |
| F ₂₄ | Mean | 7.653 × 10 ⁰ | 0.000 × 10⁰ | 5.258 × 10 ⁻¹ | 0.000 × 10⁰ | 7.259 × 10 ⁰ | 6.534 × 10 ⁰ | 1.832 × 10 ⁰ |
| | Std | 9.000 × 10 ⁰ | 0.000 × 10 ⁰ | 8.550 × 10 ⁻¹ | 0.000 × 10 ⁰ | 4.673 × 10 ⁰ | 3.388 × 10 ⁰ | 2.790 × 10 ⁰ |
| F ₂₅ | Mean | 2.261 × 10 ¹ | 5.679 × 10 ⁻² | 3.980 × 10 ⁻¹ | 3.286 × 10⁻² | 3.177 × 10 ⁰ | 3.383 × 10 ⁻¹ | 7.466 × 10 ⁻¹ |
| | Std | 5.471 × 10 ⁰ | 4.972 × 10 ⁻² | 2.258 × 10 ⁻¹⁶ | 4.553 × 10 ⁻² | 8.810 × 10 ⁻¹ | 1.214 × 10 ⁻¹ | 2.313 × 10 ⁻¹ |
| F ₂₆ | Mean | 3.129 × 10 ⁰ | 4.385 × 10 ⁻¹⁵ | 0.000 × 10⁰ | 0.000 × 10⁰ | 2.479 × 10 ⁻¹ | 0.000 × 10⁰ | 1.552 × 10 ⁻¹⁴ |
| | Std | 1.848 × 10 ⁰ | 1.532 × 10 ⁻¹⁴ | 0.000 × 10 ⁰ | 0.000 × 10 ⁰ | 4.433 × 10 ⁻¹ | 0.000 × 10 ⁰ | 8.417 × 10 ⁻¹⁶ |

Table 5. Friedman test statistical results of seven algorithms.

| Function | Sum of Squares | Degree of Freedom | Mean Squares | p-Value |
|----------|----------------|-------------------|--------------|---------------------------|
| F_1 | 826.667 | 6 | 137.778 | 1.372×10^{-35} |
| F_2 | 770.667 | 6 | 128.444 | 4.81761×10^{-33} |
| F_3 | 814.067 | 6 | 135.678 | 5.13321×10^{-35} |
| F_4 | 821.333 | 6 | 136.889 | 2.39823×10^{-35} |
| F_5 | 704.867 | 6 | 117.478 | 4.65662×10^{-30} |
| F_6 | 591.533 | 6 | 98.5889 | 6.18862×10^{-25} |
| F_7 | 690 | 6 | 115 | 3.39314×10^{-36} |
| F_8 | 821.533 | 6 | 136.922 | 1.72524×10^{-35} |
| F_9 | 836.267 | 6 | 139.378 | 5.01757×10^{-36} |
| F_{10} | 440.933 | 6 | 73.4889 | 3.53761×10^{-18} |
| F_{11} | 836.267 | 6 | 139.378 | 5.01757×10^{-36} |
| F_{12} | 730.2 | 6 | 121.7 | 3.30782×10^{-31} |
| F_{13} | 540 | 6 | 90 | 3.39314×10^{-36} |
| F_{14} | 536.233 | 6 | 89.3722 | 1.27613×10^{-33} |
| F_{15} | 737.65 | 6 | 122.942 | 4.72241×10^{-34} |
| F_{16} | 643.467 | 6 | 107.244 | 3.58476×10^{-28} |
| F_{17} | 742.083 | 6 | 123.681 | 4.75385×10^{-32} |
| F_{18} | 724.917 | 6 | 120.819 | 1.97730×10^{-33} |
| F_{19} | 648.467 | 6 | 108.078 | 1.66346×10^{-27} |
| F_{20} | 727.933 | 6 | 121.322 | 4.19129×10^{-31} |
| F_{21} | 683.467 | 6 | 113.911 | 4.33937×10^{-29} |
| F_{22} | 713.867 | 6 | 118.978 | 1.82038×10^{-30} |
| F_{23} | 562.667 | 6 | 93.7778 | 1.23609×10^{-23} |
| F_{24} | 590.25 | 6 | 98.375 | 5.55632×10^{-27} |
| F_{25} | 817.8 | 6 | 136.3 | 2.83026×10^{-35} |
| F_{26} | 719.333 | 6 | 119.889 | 8.82755×10^{-35} |

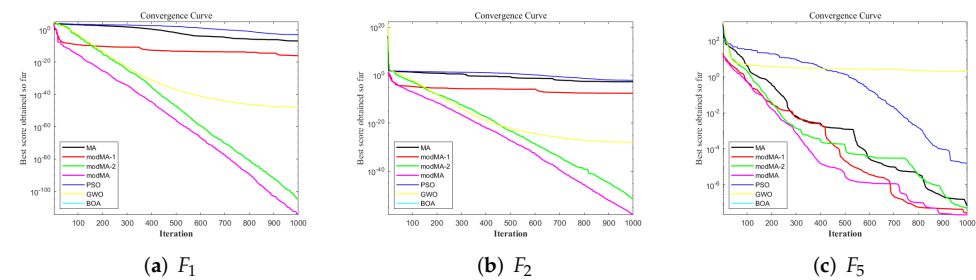


Figure 5. Comparison of convergence curve of MA, modMA-1, modMA-2, modMA, PSO, GWO and BOA in benchmark functions F_1 , F_2 and F_5 when $Dim = 50$.

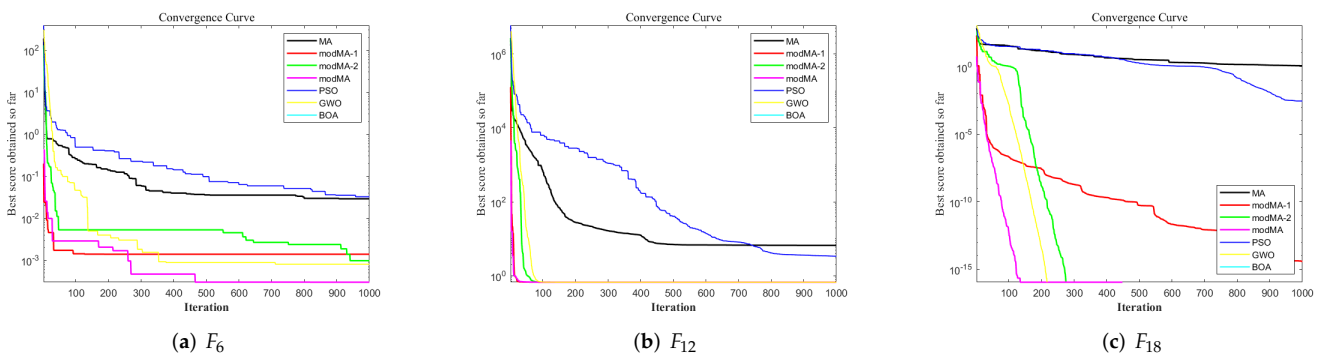


Figure 6. Cont.

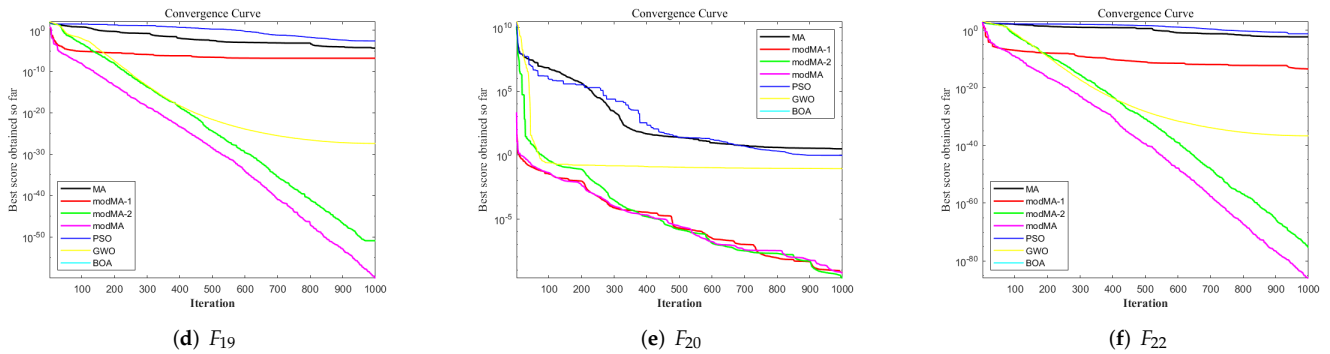


Figure 6. Comparison of convergence curve between MA, modMA-1, modMA-2, modMA, PSO, GWO and BOA with in benchmark functions $F_6, F_{12}, F_{18}, F_{19}, F_{20}$ and F_{22} Dim = 50.

Table 6. The statistical results of the Wilcoxon rank-sum test for seven algorithms.

| Function | modMA | MA | modMA-1 | modMA-2 | PSO | GWO | BOA |
|----------|-------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| F_1 | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 4.504×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_2 | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 5.494×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_3 | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 4.504×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_4 | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 7.773×10^{-9} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_5 | N/A | 3.848×10^{-3} | 2.151×10^{-2} | 3.265×10^{-2} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_6 | N/A | 3.020×10^{-11} | 1.518×10^{-3} | 1.058×10^{-3} | 3.020×10^{-11} | 3.368×10^{-4} | 5.369×10^{-2} |
| F_7 | N/A | N/A | N/A | N/A | 1.685×10^{-14} | 1.212×10^{-12} | 1.212×10^{-12} |
| F_8 | N/A | 2.113×10^{-11} | 2.113×10^{-11} | 5.561×10^{-1} | 2.113×10^{-11} | 2.113×10^{-11} | 2.113×10^{-11} |
| F_9 | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 4.975×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{10} | N/A | 6.627×10^{-1} | 3.020×10^{-11} | 1.761×10^{-1} | 1.067×10^{-7} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{11} | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{12} | N/A | 3.338×10^{-11} | 1.273×10^{-2} | 2.581×10^{-1} | 3.020×10^{-11} | 6.722×10^{-10} | 3.020×10^{-11} |
| F_{13} | N/A | N/A | N/A | N/A | 1.212×10^{-12} | N/A | 1.212×10^{-12} |
| F_{14} | N/A | 5.584×10^{-3} | N/A | N/A | 1.212×10^{-12} | N/A | 1.212×10^{-12} |
| F_{15} | N/A | 1.212×10^{-12} | 1.836×10^{-10} | 1.212×10^{-12} | 1.212×10^{-12} | 5.542×10^{-3} | 3.337×10^{-1} |
| F_{16} | N/A | 1.212×10^{-12} | 1.171×10^{-12} | 1.212×10^{-12} | 1.212×10^{-12} | 8.814×10^{-7} | 2.788×10^{-3} |
| F_{17} | N/A | 1.015×10^{-11} | 1.015×10^{-11} | 1.910×10^{-1} | 1.015×10^{-11} | 7.763×10^{-12} | 1.014×10^{-11} |
| F_{18} | N/A | 1.212×10^{-12} | 1.206×10^{-12} | 8.152×10^{-2} | 1.212×10^{-12} | N/A | 4.523×10^{-12} |
| F_{19} | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 3.690×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{20} | N/A | 3.020×10^{-11} | 2.266×10^{-3} | 5.106×10^{-1} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{21} | N/A | 3.690×10^{-11} | 2.608×10^{-2} | 1.529×10^{-5} | 7.697×10^{-4} | 5.092×10^{-8} | 3.020×10^{-11} |
| F_{22} | N/A | 3.020×10^{-11} | 3.020×10^{-11} | 4.077×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{23} | N/A | 1.501×10^{-2} | 7.062×10^{-1} | 6.627×10^{-1} | 5.555×10^{-2} | 3.020×10^{-11} | 3.020×10^{-11} |
| F_{24} | N/A | 1.212×10^{-12} | N/A | 4.777×10^{-8} | 1.212×10^{-12} | 1.212×10^{-12} | 1.455×10^{-4} |
| F_{25} | N/A | 2.772×10^{-11} | 9.762×10^{-3} | 1.182×10^{-12} | 2.930×10^{-11} | 2.964×10^{-11} | 2.964×10^{-11} |
| F_{25} | N/A | 1.212×10^{-12} | 3.428×10^{-7} | N/A | 1.212×10^{-12} | N/A | 1.167×10^{-12} |

4.4. Uav Path Planning Experiment

In this section, the proposed modMA is evaluated in two separate scenarios, the threat locations for the two cases are listed in Table 7. We select PSO [11], GWO [20], BOA [47], MA [24], modMA-1, modMA-2, and modMA for experiments. The parameter settings of these algorithms are the same as in Section 4.1.

In both cases, we assume that the start point and target point of the drone are (0, 0), and (500, 500). The maximum turning angle ϕ_k is set to 45 degrees. The values of weight coefficients ω_1 and ω_2 are set to 0.95 and 0.05, respectively. This paper discusses $D = 30$ and $D = 50$ in each scenario separately. D is set to 50, that is, the distance between the start point and target point is cut into 50 segments. To guarantee equality of the comparison, the number of particles, the pre-defined maximum iteration value, and each algorithm's number of runs are all set to 40, 200, and 30.

Table 7. Space environment settings.

| Case Number | Serial Number | Obstacle Center | Obstacle Radius |
|-------------|---------------|-----------------|-----------------|
| 1 | 1 | (50, 105) | 70 |
| | 2 | (125, 250) | 35 |
| | 3 | (304, 400) | 45 |
| | 4 | (404, 320) | 50 |
| | 5 | (440, 440) | 20 |
| | 6 | (280, 310) | 25 |
| | 7 | (230, 220) | 25 |
| | 8 | (230, 100) | 50 |
| 2 | 1 | (160, 160) | 15 |
| | 2 | (50, 105) | 70 |
| | 3 | (275, 185) | 80 |
| | 4 | (400, 425) | 40 |
| | 5 | (125, 250) | 35 |
| | 6 | (275, 325) | 28 |
| | 7 | (450, 250) | 45 |
| | 8 | (175, 410) | 70 |
| | 9 | (35, 325) | 50 |
| | 10 | (330, 300) | 25 |

The experimental results are listed in Table 8. The corresponding convergence curves are shown in Figures 7a–f and 8a–f. From Figures 7a–f and 8a–f, all the seven algorithms can find collision-free flight paths, but they perform differently in terms of path length and smoothness. It can be seen from Table 8 that in the case of $D = 30$, that is, in a lower dimension, each algorithm can achieve good results, and the path length planned by modMA is shorter and the path is smoother. In the case of $D = 50$, that is, in a higher dimension, modMA, modMA-1, and modMA-2 greatly improve the performance of MA. modMA and modMA-1 can achieve better results in most cases, and both have good initial solutions and convergence rates, but the fluctuation of modMA is smaller than that of modMA-1. Among all the comparison algorithms, the overall cost of modMA planning is the smallest in most cases, which shows that the path planned by the modMA can reach the best when the UAV avoids collision, which proves the effectiveness of the modMA.

Table 8. The average and standard deviation values of the total cost function over thirty runs.

| Case Number | D | Results | MA | modMA-1 | modMA-2 | modMA | BOA | PSO | GWO |
|-------------|----|---------|---------|---------|---------|----------------|---------|---------|---------|
| 1 | 30 | Avg | 705.432 | 690.146 | 692.234 | 689.532 | 711.309 | 698.138 | 693.970 |
| | | Std | 6.199 | 1.146 | 3.313 | 1.014 | 7.455 | 6.897 | 2.749 |
| | 50 | Avg | 753.834 | 699.756 | 723.285 | 698.312 | 720.847 | 739.353 | 710.557 |
| | | Std | 18.143 | 2.822 | 11.390 | 1.419 | 13.549 | 18.421 | 9.951 |
| 2 | 30 | Avg | 717.855 | 692.493 | 693.226 | 691.735 | 711.219 | 694.774 | 694.458 |
| | | Std | 10.718 | 1.729 | 5.087 | 1.359 | 4.180 | 3.273 | 1.675 |
| | 50 | Avg | 781.436 | 703.315 | 747.785 | 702.119 | 721.602 | 732.321 | 710.133 |
| | | Std | 28.557 | 3.123 | 23.130 | 1.978 | 6.189 | 12.167 | 7.973 |

The path planning problem has many local solutions, this makes classical optimization techniques difficult to solve. modMA shows excellent performance on this complex problem, i.e., the quality of the solution and the stability of the solution. In addition, modMA also has good initial solution and fast convergence speed. However, the model established in this paper still has some shortcomings. After collision handling, that is, the operation of moving the waypoint to the edge of the obstacle, the planned path length may be longer than the theoretical optimum.

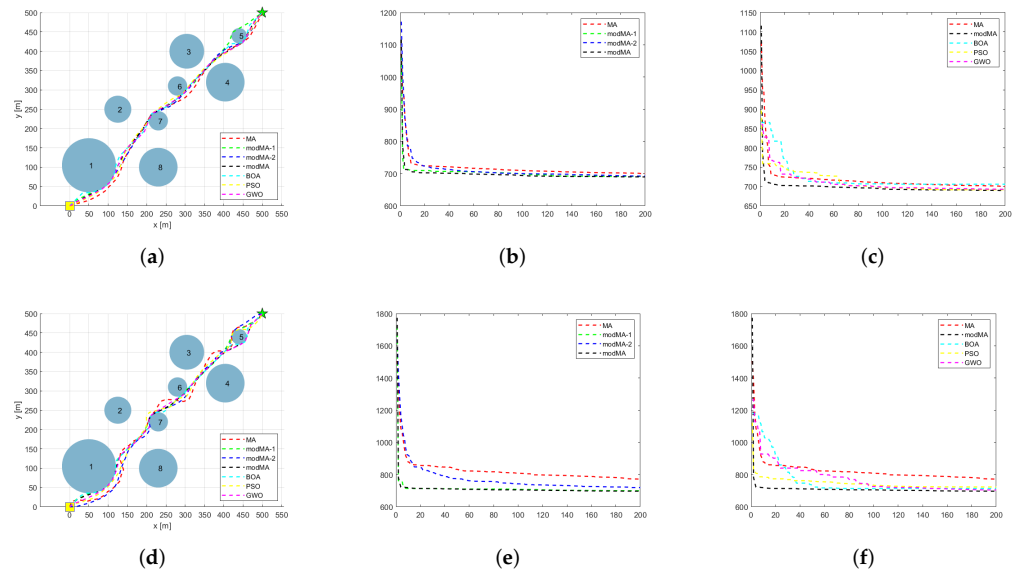


Figure 7. Path diagram of a single run in Case 1. (a) Comparative path planning results in Case 1, $Dim = 30$; (b) Evolution curves of four algorithms in Case 1, $Dim = 30$; (c) Evolution curves of different algorithms in Case 1, $Dim = 30$; (d) Comparative path planning results in Case 1, $Dim = 50$; (e) Evolution curves of four algorithms in Case 1, $Dim = 50$; (f) Evolution curves of different algorithms in Case 1, $Dim = 50$.

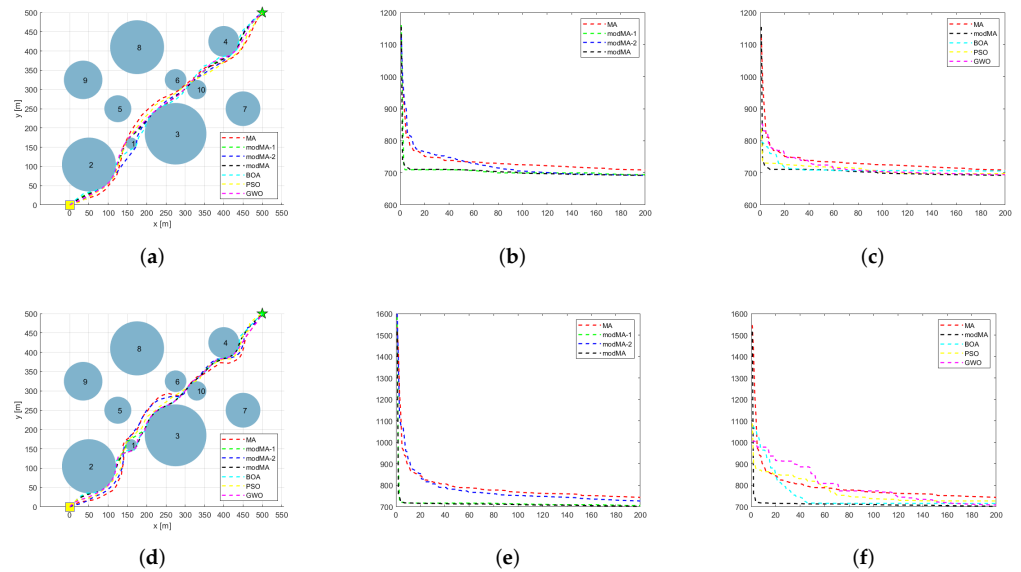


Figure 8. Path diagram of a single run in Case 2. (a) Comparative path planning results in Case 2, $Dim = 30$. (b) Evolution curves of four algorithms in Case 2, $Dim = 30$. (c) Evolution curves of different algorithms in Case 2, $Dim = 30$. (d) Comparative path planning results in Case 2, $Dim = 50$. (e) Evolution curves of four algorithms in Case 2, $Dim = 50$. (f) Evolution curves of different algorithms in Case 2, $Dim = 50$.

5. Conclusions

A new algorithm modMA which incorporates a mixed strategy consisting of three improved techniques is proposed throughout this research for solving the UAV path planning problem. By introducing the EDIW strategy to balance the particles' processes of exploration and exploitation. By perturbing the position of each male particle with an adaptive Cauchy mutation operator, the diversity of the swarm is increased. By introducing an enhanced crossover operator, the exploration ability of the modMA is enhanced. To

assess the efficacy of these three improvement techniques on MA, two experiments are designed. The suggested modMA is tested on 26 benchmark functions, and it has obvious advantages in finding both the best value and convergence speed. Finally, the proposed modMA, together with PSO, GWO, and BOA, are applied to the UAV path planning problem; simulation experiments show that the effect of modMA which is applied to the UAV path planning problem is relatively stable, and the quality of the planned path is high. The time cost of modMA in the path planning problem of UAV is higher than that of PSO, BOA and GWO, which is determined by its algorithm structure, but the convergence speed is fast in the process of solving the problem.

In future work, hybrid [49,50] can be introduced to further enhance the performance of the modMA. The cost of the drone collision threat can also be considered in the UAV path planning problem.

Author Contributions: Conceptualization, X.W. and J.-S.P.; Formal analysis, X.W., J.-S.P., Q.Y. and S.-C.C.; Methodology, X.W., J.-S.P., L.K., V.S. and S.-C.C.; Validation, J.-S.P., Q.Y., L.K. and V.S.; Writing—original draft, X.W.; Writing—review & editing, X.W., J.-S.P., Q.Y., L.K., V.S. and S.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Table A1. Unimodal benchmark functions.

| Name | Formula of Functions | Dim | Range | f_{min} |
|---------------|--|-----|---------------|-----------|
| Sphere | $F_1(x) = \sum_{i=1}^{Dim} x_i^2$ | 50 | [−100, 100] | 0 |
| Schwefel 2.22 | $F_2(x) = \sum_{i=1}^{Dim} x_i + \prod_{i=1}^{Dim} x_i $ | 50 | [−10, 10] | 0 |
| Schwefel 1.2 | $F_3(x) = \sum_{i=1}^{Dim} \left(\sum_{j=i}^{Dim} x_j \right)^2$ | 50 | [−100, 100] | 0 |
| Schwefel 2.21 | $F_4(x) = \max\{ x_i , 1 \leq i \leq Dim\}$ | 50 | [−10, 10] | 0 |
| Step | $F_5(x) = \sum_{i=1}^{Dim} (x_i + 0.5)^2$ | 50 | [−10, 10] | 0 |
| Quartic | $F_6(x) = \sum_{i=1}^{Dim} Dim \cdot x_i^2 + rand(0, 1)$ | 50 | [−1.28, 1.28] | 0 |
| Exponential | $F_7(x) = \exp\left(0.5 \sum_{i=1}^{Dim} x_i\right)$ | 50 | [−10, 10] | 0 |
| Sum power | $F_8(x) = \sum_{i=1}^{Dim} x_i ^{(i+1)}$ | 50 | [−1, 1] | 0 |
| Sum square | $F_9(x) = \sum_{i=1}^{Dim} (Dim \cdot x_i^2)$ | 50 | [−10, 10] | 0 |
| Rosenbrock | $F_{10}(x) = \sum_{i=1}^{Dim} \left(100(x_{i+1} - x_i^2) + (x_i - 1)^2\right)$ | 50 | [−5, 10] | 0 |
| Zakharov | $F_{11}(x) = \sum_{i=1}^{Dim} x_i^2 + \left(\sum_{i=1}^{Dim} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{Dim} 0.5ix_i\right)^4$ | 50 | [−5, 10] | 0 |
| Trid | $F_{12}(x) = (x_i - 1)^2 + \sum_{i=1}^{Dim} i \cdot (2x_i^2 - x_{i-1})^2$ | 50 | [−10, 10] | 0 |
| Elliptic | $F_{13}(x) = \sum_{i=1}^{Dim} (10^6)^{(i-1)/(Dim-1)} \cdot x_i^2$ | 50 | [−100, 100] | 0 |
| Cigar | $F_{14}(x) = x_1^2 + 10^6 \sum_{i=2}^{Dim} x_i^2$ | 50 | [−100, 100] | 0 |

Table A2. Multimodal benchmark functions.

| Name | Formula of Functions | Dim | Range | f_{\min} |
|-------------|--|-----|---------------|------------|
| Rastrigin | $F_{15}(x) = \sum_{i=1}^{Dim} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | 50 | [-5.12, 5.12] | 0 |
| NCRastrigin | $F_{16}(x) = \sum_{i=1}^{Dim} [y_i^2 - 10 \cos(2\pi y_i) + 10],$ $y_i = \begin{cases} x_i, & x_i < 0.5 \\ \text{round}(2x_i)/2, & x_i > 0.5 \end{cases}$ | 50 | [-5.12, 5.12] | 0 |
| Ackley | $F_{17}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{Dim} \sum_{i=1}^{Dim} x_i^2}\right)$ $+ \exp\left(\frac{1}{Dim} \sum_{i=1}^{Dim} \cos(2\pi x_i)\right) + 20 + \exp(1)$ | 50 | [-50, 50] | 0 |
| Griewank | $F_{18}(x) = \frac{1}{4000} \sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | 50 | [-600, 600] | 0 |
| Alpine | $F_{19}(x) = \sum_{i=1}^{Dim} x_i \cdot \sin(x_i) + 0.1x_i $ | 50 | [-10, 10] | 0 |
| Penalized 1 | $F_{20}(x) = \frac{\pi}{Dim} \left\{ \sum_{i=1}^{Dim-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_1)] + (y_{Dim} - 1)^2 + 10 \sin^2(\pi y_1) \right\} + \sum_{i=1}^{Dim} u(x_i, 10, 100, 4)$ $y_i = 1 + (x_i + 1)/4, u_{y_i, a, k, m} = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$ | 50 | [-100, 100] | 0 |
| Penalized 2 | $F_{21}(x) = \frac{1}{10} \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{Dim-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_{Dim-1})^2 (1 + \sin^2(2\pi x_{i+1})) \right\} + \sum_{i=1}^{Dim} u(x_i, 5, 100, 4)$ | 50 | [-100, 100] | 0 |
| Schwefel | $F_{22}(x) = \sum_{i=1}^{Dim} x_i \cdot \sin(\sqrt{ x_i }) $ | 50 | [-100, 100] | 0 |
| Levy | $F_{23}(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{Dim-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + x_{Dim} - 1 \cdot [1 + \sin^2(2\pi x_{Dim})]$ | 50 | [-10, 10] | 0 |
| Weierstrass | $F_{24}(x) = \sum_{i=1}^{Dim} \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - Dim \cdot \sum_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)], a = 0.5, b = 3, k_{\max} = 20$ | 50 | [-1, 1] | 0 |
| Solomon | $F_{25}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^{Dim} x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^{Dim} x_i^2}$ | 50 | [-100, 100] | 0 |
| Bohachevsky | $F_{26}(x) = \sum_{i=1}^{Dim} [x_i^2 + 2x_{i+1}^2 - 0.3 \cdot \cos(3\pi x_i)]$ | 50 | [-10, 10] | 0 |

References

1. Yu, X.; Zhang, Y. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Prog. Aerosp. Sci.* **2015**, *74*, 152–166. [[CrossRef](#)]
2. Huo, L.; Zhu, J.; Wu, G.; Li, Z. A novel simulated annealing based strategy for balanced UAV task assignment and path planning. *Sensors* **2020**, *20*, 4769. [[CrossRef](#)]
3. Dhulkefl, E.J.; Durdu, A. Path planning algorithms for unmanned aerial vehicles. *Int. J. Trend Sci. Res. Dev.* **2019**, *3*, 359–362. [[CrossRef](#)]
4. Zhang, Z.; Wu, J.; Dai, J.; He, C. A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment. *IEEE Access* **2020**, *8*, 122757–122771. [[CrossRef](#)]
5. Zhang, D.; Xu, Y.; Yao, X. An improved path planning algorithm for unmanned aerial vehicle based on rrt-connect. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 4854–4858.
6. Huo, L.; Zhu, J.; Li, Z.; Ma, M. A hybrid differential symbiotic organisms search algorithm for UAV path planning. *Sensors* **2021**, *21*, 3037. [[CrossRef](#)]
7. Zhang, D.; Duan, H. Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning. *Neurocomputing* **2018**, *313*, 229–246. [[CrossRef](#)]
8. Zhang, X.; Duan, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. *Appl. Soft Comput.* **2015**, *26*, 270–284. [[CrossRef](#)]
9. Rajabioun, R. Cuckoo optimization algorithm. *Appl. Soft Comput.* **2011**, *11*, 5508–5518. [[CrossRef](#)]
10. Song, P.C.; Pan, J.S.; Chu, S.C. A parallel compact cuckoo search algorithm for three-dimensional path planning. *Appl. Soft Comput.* **2020**, *94*, 106443. [[CrossRef](#)]
11. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, USA, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
12. Chu, S.C.; Roddick, J.F.; Pan, J.S. A parallel particle swarm optimization algorithm with communication strategies. *J. Inf. Sci. Eng.* **2005**, *21*, 809–818.
13. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [[CrossRef](#)]
14. Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.
15. Roberge, V.; Tarbouchi, M.; Labonté, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Info.* **2012**, *9*, 132–141. [[CrossRef](#)]
16. Basturk, B. An artificial bee colony (ABC) algorithm for numeric function optimization. In Proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 12–14 May 2006.
17. Lei, L.; Shiru, Q. Path planning for unmanned air vehicles using an improved artificial bee colony algorithm. In Proceedings of the 31st Chinese Control Conference, Hefei, China, 25–27 July 2012; pp. 2486–2491.
18. Price, K.V. Differential evolution: A fast and simple numerical optimizer. In Proceedings of the North American Fuzzy Information Processing, Berkeley, CA, USA, 19–22 June 1996; pp. 524–527.
19. Pan, J.S.; Liu, N.; Chu, S.C. A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning. *IEEE Access* **2020**, *8*, 17691–17712. [[CrossRef](#)]
20. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
21. Lv, J.X.; Yan, L.J.; Chu, S.C.; Cai, Z.M.; Pan, J.S.; He, X.K.; Xue, J.K. A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3D path planning of multiple UAVs in power inspection. *Neural Comput. Appl.* **2022**, *193*, 509–532. [[CrossRef](#)]
22. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [[CrossRef](#)]
23. Konatowski, S.; Pawłowski, P. Ant colony optimization algorithm for UAV path planning. In Proceedings of the 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 20–24 February 2018; pp. 177–182.
24. Zervoudakis, K.; Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **2020**, *145*, 106559. [[CrossRef](#)]
25. Kennedy, J. Bare bones particle swarms. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS'03 (Cat. No. 03EX706), Indianapolis, IN, USA, 26 April 2003; pp. 80–87.
26. Juan, Z.; Zheng-Ming, G. Bare bones mayfly optimization algorithm. In Proceedings of the 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 23–25 October 2020; pp. 238–241.
27. Shaheen, M.A.; Hasanien, H.M.; El Moursi, M.; El-Fergany, A.A. Precise modeling of PEM fuel cell using improved chaotic MayFly optimization algorithm. *Int. J. Energy Res.* **2021**, *45*, 18754–18769. [[CrossRef](#)]
28. Gigras, Y.; Gupta, K.; Choudhury, K. A comparison between bat algorithm and cuckoo search for path planning. *Int. J. Innov. Res. Comput. Commun. Eng.* **2015**, *3*, 4459–4466.
29. Zhang, S.; Zhou, Y.; Li, Z.; Pan, W. Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Adv. Eng. Softw.* **2016**, *99*, 121–136. [[CrossRef](#)]

30. Pan, J.S.; Liu, J.L.; Hsiung, S.C. Chaotic cuckoo search algorithm for solving unmanned combat aerial vehicle path planning problems. In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, Zhuhai, China, 22–24 February 2019; pp. 224–230.
31. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [[CrossRef](#)]
32. Wang, G.; Guo, L.; Duan, H.; Liu, L.; Wang, H. A modified firefly algorithm for UCAV path planning. *Int. J. Hybrid Inf. Technol.* **2012**, *5*, 123–144.
33. Zhu, W.; Duan, H. Chaotic predator–prey biogeography-based optimization approach for UCAV path planning. *Aerosp. Sci. Technol.* **2014**, *32*, 153–161. [[CrossRef](#)]
34. Yang, X.S. Firefly algorithm, Levy flights and global optimization. In *Research and Development in Intelligent Systems XXVI*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 209–218.
35. Foo, J.L.; Knutzon, J.; Kalivarapu, V.; Oliver, J.; Winer, E. Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization. *J. Aerosp. Comput. Inf. Commun.* **2009**, *6*, 271–290. [[CrossRef](#)]
36. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
37. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [[CrossRef](#)]
38. Gao, Y.L.; An, X.H.; Liu, J.M. A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In Proceedings of the 2008 International Conference on Computational Intelligence and Security, Washington, DC, USA, 13–17 December 2008; Volume 1, pp. 61–65.
39. Rathore, A.; Sharma, H. Review on inertia weight strategies for particle swarm optimization. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 76–86.
40. Cao, Z.; Shi, Y.; Rong, X.; Liu, B.; Du, Z.; Yang, B. Random grouping brain storm optimization algorithm with a new dynamically changing step size. In *Proceedings of the International Conference in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 357–364.
41. Zou, Y.; Liu, P.X.; Yang, C.; Li, C.; Cheng, Q. Collision detection for virtual environment using particle swarm optimization with adaptive cauchy mutation. *Clust. Comput.* **2017**, *20*, 1765–1774. [[CrossRef](#)]
42. Chakraborty, F.; Roy, P.K.; Nandi, D. Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding. *Evol. Intell.* **2019**, *12*, 445–467. [[CrossRef](#)]
43. Fogarty, T.C. Varying the Probability of Mutation in the Genetic Algorithm. In Proceedings of the 3rd International Conference on Genetic Algorithms, Fairfax, VA, USA, 4–7 June 1989; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1989; pp. 104–109.
44. Lin, W.Y.; Lee, W.Y.; Hong, T.P. Adapting Crossover and Mutation Rates in Genetic Algorithms. *J. Inf. Sci. Eng.* **2003**, *19*, 889–903.
45. Van Laarhoven, P.J.; Aarts, E.H. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 1987; pp. 7–15.
46. Meng, A.b.; Chen, Y.c.; Yin, H.; Chen, S.Z. Crisscross optimization algorithm and its application. *Knowl. Based Syst.* **2014**, *67*, 218–229. [[CrossRef](#)]
47. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [[CrossRef](#)]
48. Zhang, M.; Long, D.; Qin, T.; Yang, J. A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems. *Symmetry* **2020**, *12*, 1800. [[CrossRef](#)]
49. Wang, G.G.; Gandomi, A.H.; Zhao, X.; Chu, H.C.E. Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Comput.* **2016**, *20*, 273–285. [[CrossRef](#)]
50. Pan, J.S.; Zhuang, J.; Liao, L.; Chu, S.C. Advanced equilibrium optimizer for electric vehicle routing problem with time windows. *J. Netw. Intell.* **2021**, *6*, 216–237.