ITC 2/53

Information Technology and Control
Vol. 53 / No. 2 / 2024
pp. 372-389
DOI 10.5755/j01.itc.53.2.35955

Multi-strategy Improved Pelican Optimization Algorithm for Mobile Robot Path Planning

Received 2024/01/05 | Accepted after revision 2024/02/29

HOW TO CITE: Li, C. Q., Jiang, Z. F., Huang, Y. P. (2024). Multi-strategy Improved Pelican Optimization Algorithm for Mobile Robot Path Planning. *Information Technology and Control*, 53(2), 372-389. https://doi.org/10.5755/j01.itc.53.2.35955

Multi-strategy Improved Pelican Optimization Algorithm for Mobile Robot Path Planning

Chun Qing Li, Zheng Feng Jiang

College of Mathematics, Physics and Electronic Information Engineering, Guangxi Minzu Normal University, Chongzuo 532200, China; e-mails: 516260129@qq.com (C. Q. Li); 305909010@qq.com (Z. F. Jiang)

Yong Ping Huang

College of Mathematics, Physics and Electronic Information Engineering, Guangxi Minzu Normal University, Chongzuo 532200, China; e-mail: 272358014@qq.com

Corresponding author: ZhengFeng Jiang, 305909010@qq.com

In response to the problems of easily falling into local optima, low path planning accuracy, and slow convergence speed when applying the traditional pelican optimization algorithm to the mobile robot path planning problem, a multi-strategy improved pelican optimization algorithm (MPOA) is proposed. In the initialization stage, chaotic mapping is used to increase the diversity of the pelican population individuals. In the exploration stage, an adaptive feedback adjustment factor is proposed to adjust the local optima of pelican individuals' positions and balance the algorithm's local development capability. In the development stage, the Lévy flight strategy is introduced to adjust the domain radius of the pelican population individuals, and the Gaussian mutation mechanism is used to disturb individuals that have fallen into local optima. Simulation experimental results show that the improved algorithm has significantly improved and effectively shortened the length of the planned path.

KEYWORDS: Pelican Optimization Algorithm, mobile robot, path planning, multi strategy improved.

1. Introduction

Mobile robot path planning has consistently remained a significant area of research within the field of robotics. The primary objective of this research is to identify the most efficient route from the initial point to the desired destination across a range of different map environments. This particular aspect of robot navigation and autonomous movement is of utmost importance and finds extensive applications in diverse fields including industrial automation, unmanned vehicles, and logistics distribution. In recent years, with the rapid development of bio-inspired intelligent optimization algorithms, scholars have conducted a large amount of research on improving intelligent optimization algorithms applied to robot path planning. Mainly, there are ant colony and its improved algorithms [4-5], particle swarm and its improved algorithms [10, 13], genetic algorithm and its improved algorithms [2, 12], etc. In addition, with the introduction of new bio-inspired intelligent algorithms, an increasing number of intelligent optimization algorithms are being applied to the research of robot path optimization, such as whale optimization algorithm [17], sparrow search algorithm [7], artificial fish swarm optimization algorithm [8], albatross algorithm [6], etc.

Based on the hunting behavior of pelicans in the natural world, Pavel Trojovský and Mohammad Dehghani [11] proposed a novel bio-inspired intelligent algorithm called the Pelican Optimization Algorithm (POA) in 2022. This algorithm considers candidate solutions as the positions of pelicans and the objective function values as the quality of food, aiming to find the optimal solution by simulating the foraging behavior of pelicans. The basic steps of the Pelican Optimization Algorithm are as follows:

Step 1: Initialize the population by randomly generating a set of initial solutions as the population.

Step 2: Evaluate fitness by calculating the fitness value of each individual, which is the value of the objective function.

Step 3: Update positions by using the fitness value of the current individual and neighborhood information to update the position of each individual.

Step 4: Update fitness by recalculating the fitness value of each individual based on the new positions.

Step 5: Determine the termination condition. If the termination condition is met (e.g., reaching the maximum number of iterations or finding a satisfactory solution), stop the algorithm; otherwise, go back to Step 3.

By iteratively updating the positions and fitness of individuals, the Pelican Optimization Algorithm can gradually improve the quality of solutions, eventually finding the optimal solution or approaching it. This algorithm has potential applications in solving optimization problems and can be adapted to different problem domains by adjusting parameters and improving algorithm details.

As a result, researchers have successfully applied the POA to various engineering problems, including network attack detection models [1], image issues [13-14], and asynchronous motor fault diagnosis [3]. However, when dealing with more complex engineering problems such as robot path planning, the algorithm still faces challenges such as falling into local optima, low accuracy, and slow convergence speed.

In the literature [15], the pelican optimization algorithm (POA) was first applied to the mobile robot path planning problem and demonstrated the feasibility of the POA algorithm in robot path planning. The algorithm can obtain shorter movement paths in a shorter period of time. However, for more complex engineering problems, the algorithm still faces some challenges and limitations, including sensitivity to parameter selection and susceptibility to local optima. Therefore, in practical applications, adjustments and improvements need to be made based on specific problems to overcome these challenges and limitations

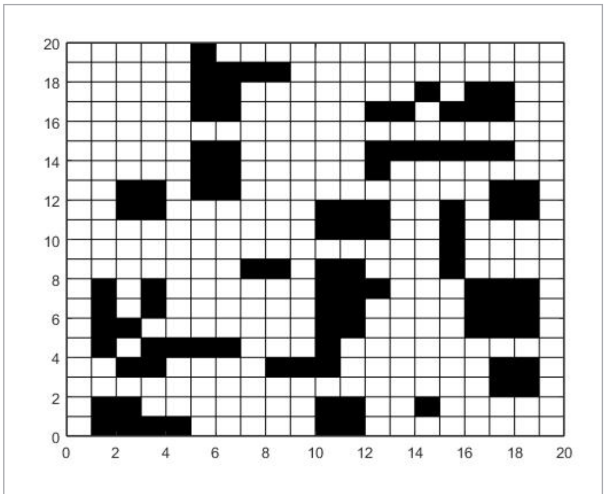
This paper proposes a multi-strategy improved Pelican optimization algorithm (MPOA). The algorithm uses the Cubic chaotic mapping to initialize the positions of the pelican population, aiming to improve the randomness issue of randomly generated populations in traditional POA algorithms and enhance the diversity of the population. Additionally, an adaptive feedback adjustment factor w is introduced to ensure that the updates of the pelican positions are within a certain range, thereby addressing the problem of blind trend-following in the algorithm's late-stage local development after exploration. During the development stage of the algorithm, Levy flight strategy is first employed to update the positions of the pelican individuals, maintaining their global optimization capability. Then, a Gaussian mutation mechanism is introduced for the later updates of the pelican positions. Through this combination of multiple strategies, the MPOA algorithm is able to better balance the abilities of global search and local search, thus improving the algorithm's performance. To evaluate the performance of the proposed improved algorithm, experiments were conducted using the benchmark test suite CEC 2017. The experimental results demonstrate that the MPOA algorithm performs well in this test suite. Furthermore, the improved algorithm proposed in this paper was applied in simulation experiments for mobile robot path planning. The simulation results show that the

MPOA algorithm can accelerate convergence speed, reduce path length, and improve work efficiency. In conclusion, the multi-strategy improved Pelican optimization algorithm (MPOA) proposed in this paper demonstrates good performance in benchmark testing and mobile robot path planning, and has the potential to play an important role in practical applications.

2. Problem Model

This paper models the working environment of mobile robots as a structurally simple, accurate, and reliable grid map environment. The robot's working environment can be simplified as shown in Figure 1, which consists of a certain number of identical gridded areas. In the figure, black represents obstacles and is denoted by 1, while white represents freely passable grids and is denoted by 0. The research content of this paper is how the robot selects the freely passable white grids in the map model to obtain the shortest path from the starting point to the end point.

Figure 1
Map model



3. Basic Pelican Optimization Algorithm Principles

In the POA algorithm, the behavior and strategy of pelicans in attacking and hunting are simulated to update candidate solutions. The hunting process is divided into two stages: the exploration stage and the exploitation stage.

3.1. Initialization

In the POA algorithm, every individual within the population serves as a potential solution, and each member is responsible for computing the value of the optimization problem variables based on its position in the search space. Initially, the variable values are randomly initialized within the upper and lower bounds according to the problem using the Equation, as depicted in Equation (1) for the mathematical representation of population initialization.

$$x_{i,j} = l_j + rand.(u_j - l_j), i = 1,2,\dots,N, j = 1,2,\dots,m, \quad (1)$$

$x_{i,j}$ represents the position of the jth dimension of the ith pelican, where N denotes the number of pelican populations and m signifies the dimension of the problem to be solved. The variable *rand* denotes a random number within the range of [0,1], while u_j and l_j represent the upper and lower bounds of the jth dimension of the problem.

A's population members of the pelican species use the matrix in Equation (2) to represent each candidate solution, with each row representing a candidate solution and the columns in the matrix representing the values of problem variables.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,1} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}. \quad (2)$$

The matrix X denotes the population distribution of pelicans, while the variable X_i signifies the spatial coordinates of the individual ith pelican.

The POA algorithm utilizes the objective function to compute the objective function value of the pelican population. This value can be denoted by the objective function value vector Equation (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}. \quad (3)$$

F represents the objective function vector of the pelican population; F_i represents the objective function value of the ith pelican.

3.2. Exploration Stages

During the exploration phase, the heron first locates its prey and then advances towards the identified area. By emulating the heron's approach to its prey, the POA can survey the search space and leverage its exploration capability to uncover various regions within it. A key aspect of the POA is the random generation of the prey's location in the search space, which enhances the algorithm's ability to explore and solve specific search problems accurately. Equation (4) illustrates the aforementioned concepts and presents a mathematical simulation of the heron's movement towards the prey location.

$$x_{i,j}^{P_1} = \begin{cases} x_{i,j} + rand.(p_j - I.x_{i,j}), F_p < F_i; \\ x_{i,j} + rand.(x_{i,j} - p_j), \quad else, \end{cases} \tag{4}$$

$x_{i,j}^{P_1}$ represents the position of the jth dimension of the ith pelican after the first stage update; $rand$ is a random number within the range of [0,1]; I is a random integer of 1 or 2; p_j represents the position of dimension j of the prey; F_p represents the objective function value of the prey.

In the POA algorithm, the new position of the pelican is accepted on the premise that the objective function value is improved at that position, which is called effective updating and cannot move to a non-optimal region. Use mathematical Equation (5) to simulate this process.

$$X_i = \begin{cases} X_i^{P_1}, & F_i^{P_1} < F_i \\ X_i, & else \end{cases} \tag{5}$$

$X_i^{P_1}$ represents the new position of the ith pelican; $F_i^{P_1}$ is the objective function value based on the updated new position of the ith pelican after the first stage.

3.3. Development Phase

During the developmental stage, as pelicans approach the water surface, they extend their wings, lift the prey upwards, and then deposit it into their throat pouch. This surface-flying technique employed by pelicans enables them to capture a greater number of fish within their attack range. By modeling the behavioral process of pelicans, the POA algorithm can be guided to converge to a more favorable position within the hunting area, thereby enhancing the algorithm's local search and developmental capabilities. From a mathematical standpoint, the algorithm needs to explore points in close proximity to the pelican's position in order to converge to an optimal position and achieve a superior solution. The mathematical representation of the pelican's developmental stage behavior is depicted in Equation (6).

$$x_{i,j}^{P_2} = x_{i,j} + R.(1 - \tfrac{t}{T}).(2.rand - 1).x_{i,j}, \tag{6}$$

$x_{i,j}^{P_2}$ represents the position of the jth dimension of the ith pelican after the second stage update; $rand$ is a random number in the range of [0,1]; R is a random integer of 0 or 2; t is the current iteration number; T is the maximum number of iterations.

At this point, effective updates are still being employed to evaluate and decide on the acceptance or rejection of new positions of pelicans, as illustrated in Equation (7).

$$X_i = \begin{cases} X_i^{P_2}, & F_i^{P_2} < F_i \\ X_i, & else \end{cases} \tag{7}$$

4. Improved Pelican Optimization Algorithm

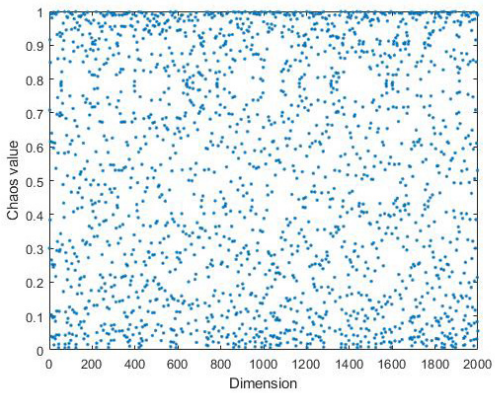
4.1. Cubic Chaotic Mapping

Chaos is a relatively common phenomenon in nonlinear systems. The values of the Cubic mapping sequence range between (0,1) , and the chaotic variables generated when ρ=2.595 has better traversability. Figure 2 shows the distribution of the Cubic mapping sequence after 2000 iterations.

In this study, we adopted the Cubic chaotic mapping to initialize the pelican population in order to enhance population diversity and ensure an even distribution. Traditional algorithms suffer from random distribution issues within the population. By using the Cubic chaotic mapping to initialize the pelican population, we can improve this problem and achieve a more evenly distributed population, thereby enhancing the initialization performance of the algorithm.

Figure 2

Distribution of Cubic Mapping Sequence Values



The Cubic chaotic mapping is a nonlinear dynamical system with highly chaotic properties. By utilizing the Cubic chaotic mapping to initialize the pelican population, we can introduce more randomness and diversity into the population. This is because the Cubic chaotic mapping possesses a large iteration range and complex nonlinear characteristics, capable of generating highly random numerical sequences.

By applying the Cubic chaotic mapping to the initialization process of the pelican population, we can ensure a more uniform distribution of individuals within the search space. This helps to avoid the issue of individuals clustering in specific areas within the population, thereby improving the global search capability of the algorithm.

This paper presents the Cubic chaotic initialization method for the pelican population, with the Equation (8) illustrating the calculation of individuals' positions within the population.

$$X_{i+1} = \rho \times X_i (1 - X_i^2), \tag{8}$$

In the Equation: ρ is the control parameter; X_i represents the position of the ith pelican.

4.2. Adaptive Feedback Control Factor

In the later stage of the MPO algorithm exploration phase, after the pelicans determine the prey's location, they search towards the prey area. However, as the pelicans get closer to the target, it may cause the algorithm to fall into a local optimum, thus unable to find the global optimum. To improve this situation, we propose an adaptive feedback adjustment factor W to enhance the algorithm. In the first phase of the POA algorithm, an adaptive inertia weight Equation is introduced as shown in Equation (9).

$$W = \begin{cases} W_{\min} - (W_{\max} - W_{\min}) \times \frac{f - f_{\min}}{f_{avg} - f_{\min}}, f \le f_{avg} \\ W_{\max} \qquad\qquad ,other \end{cases} \tag{9}$$

W is the adaptive inertia weight; $W_{\max}$ and $W_{\min}$ are the maximum and minimum values, which are set to 0.9 and 0.4, respectively; f is the fitness value; $f_{\min}$ is the optimal fitness value; f_{avg} is the average fitness value.

By introducing the adaptive inertia weight formula, we can dynamically adjust the inertia weight based on the current iteration number. In the early stages of the algorithm, the inertia weight is larger, which can help the pelicans explore the search space better, thus avoiding falling into a local optimum. As the iteration number increases, the inertia weight gradually decreases, making the individuals more inclined towards the current optimal solution, thereby improving the algorithm's convergence.

Introducing the adaptive inertia weight W into the pelican position update Equation (4), the new position update Equation is obtained as shown in Equation (10).

$$x_{i,j}^{P_1} = \begin{cases} x_{i,j} + rand \times W.(p_j - I.x_{i,j}), F_p < F_i; \\ x_{i,j} + rand \times W.(x_{i,j} - p_j), \qquad else, \end{cases} \tag{10}$$

In the Equation, all parameters have the same meanings as above.

4.3. Levy Flight Strategy

The Levy flight strategy is a random walk strategy obeying the Levy distribution, and its walking step length can achieve a larger range when searching in an unknown range area, thereby enhancing the global search capability. In practical applications, the Mantegn[a] [9] algorithm is usually used to simulate Levy's flight, and the calculation method of the step length s is as shown in Equation (11).

$$s = \frac{\mu}{|\upsilon|^{\frac{1}{\gamma}}} \tag{11}$$

Among them, μ and υ follow a normal distribution and satisfy the following conditions:

$$\mu \sim N\left(0,\sigma_\mu^2\right), \quad \upsilon \sim N(0,\sigma_\upsilon^2),$$
$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta).\sin(\frac{\pi\beta}{2})}{\Gamma\left[\frac{(1+\beta)}{2}\right]\beta.2^{\frac{(\beta-1)}{2}}} \right\}^{1/\beta} \tag{12}$$

Γ represents the gamma function; β is usually taken within the range of [0, 2], and here it is taken as 1.5; σ_υ is taken as 1.

In the second stage of the MPO algorithm, with the increase of the number of iterations, the value of coefficient $R.(1-\frac{t}{T})$ decreases, which makes the territorial radius of the pelican population smaller. This not only improves the accuracy of the scanning area but also reduces the scanning area. The population individuals continuously converge to better solutions, making it easy to fall into local optima. In order to expand the scanning area, the Lévy flight strategy is introduced into the position update Equation, as shown in Equation (13).

$$x_{i,j}^{P_2} = x_{i,j} + R.(1-\tfrac{t}{T}).(2.rand-1).x_{i,j} + 0.2 \times Levy(x_{i,j}, x_{best}), \tag{13}$$

x_{best} represents the current optimal position of the pelican, and the meanings of other parameters are the same as Equation (6).

Based on the position update Equation integrating the Levy flight strategy, the individuals of the pelican population can maintain excellent global search ability in the later stage of the algorithm. At the same time, the slow convergence speed of the Levy flight strategy is compensated by dynamically adjusting the step size of the pelican's position update.

4.4. Gaussian Mutation Mechanism

As the number of iterations increases, the pelican population continues to converge towards smaller radius areas in the domain. The Lévy flight strategy introduced in Section 2.3 helps the pelicans to some extent to escape from local optima and improve global search capability. The step size of the Lévy flight strategy is random, alternating between short-distance search and occasional longer-distance walks. Therefore, when the generated step size is short, the pelican population tends to concentrate on searching in short-distance areas, making it susceptible to blindly following a certain local optimal value. If it is found that an individual has fallen into a local optimum and cannot escape to search for a better value in other domains, a Gaussian mutation mechanism guided by the optimal solution is introduced for disturbance. Gaussian mutation is an optimization strategy that uses random numbers following a normal distribution to act on the original position vector to generate new positions, equivalent to performing domain search within a small range. In the Gaussian mutation, the current global optimal solution value is introduced to achieve information sharing between the population individuals and the current optimal solution, and to update the pelican's position through Gaussian mutation processing as shown in Equation (14).

$$x_{i,j} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x_{best}-x_{i,j})}{2\sigma^2}} x_{i,j} \tag{14}$$

σ is the Gaussian mutation parameter, with a value of 0.1; x_{best} is the current optimal position of albatross (current global optimal solution).

4.5. MPOA Algorithm Description

MPOA algorithm idea: Based on the traditional POA algorithm, the Cubic chaotic mapping is applied to the population initialization to improve the diversity of the population; in the later stage of the exploration phase, an adaptive feedback regulation factor W is introduced to regulate the problem of blindly following the local development and falling into the problem of local optimal solution in the later stage of the algorithm; the Lévy flight strategy is introduced to update the individual position calculation Equation of the pelican in the later stage of the development, broadening the search domain; using the Gaussian mutation mechanism obeying the normal distribution to achieve information sharing between the population individuals and the optimal pelican position, and perturbing the update of the individual position vector. The specific implementation steps of the algorithm are as follows:

Step 1: Set the number of pelican population N, the maximum iteration times T, the Gaussian mutation parameter σ, the Lévy flight strategy parameter β, and calculate the parameters $W_{\max}$ and $W_{\min}$.

Step 2: Use the Cubic Chaotic Mapping Strategy Equation (8) to initialize the pelican population individual positions, and calculate the pelican individual objective function value according to Equation (3).

Step 3: Calculate the adaptive feedback control factor W value according to Equation (9), and update the pelican position according to Equation (10).

Step 4: Calculate the objective function value of the pelican individual according to Equation (3). If the new objective function value is better than the current one, replace it and set the optimal objective function value as the current global optimal value x_{best}.

Step 5: Update the position of the pelican individuals according to the updated Equation (13) incorporating the Levy flight strategy. Calculate the objective function value of the pelican individual based on Equation (3). If the new objective function value is better than the current one, replace it and set the optimal objective function value as the current global optimal value x_{best}.

Step 6: Update the position of the pelican individuals according to the update Equation (14) incorporating the Gaussian mutation mechanism. Calculate the objective function value of the pelican individuals according to Equation (3). If the new objective function value is better than the current one, replace it, and set the optimal objective function value as the current global optimal value x_{best}.

Step 7: Check if the algorithm iteration has concluded. If the maximum number of iterations has been reached, then return the global optimal solution x_{best}; otherwise, proceed to step 3 and continue the iterative process.

5. Experiment and Analysis

5.1. Experimental Environment

In order to verify the effectiveness and feasibility of the proposed MPOA algorithm in the CEC2017 test suite and mobile robot path planning, we conducted simulation experiments using MATLAB R2021b. The software and hardware environment for the experiments are as follows:

Operating System: Windows 10 Professional 64-bit

Processor: 11th Gen Intel® Core™ i7-11700, with a frequency of 2.50GHz

Memory: 32GB

We will conduct the experiments in this environment to evaluate the performance of the MPOA algorithm in solving the CEC2017 test suite and mobile robot path planning problems. Through these experiments, we hope to validate the effectiveness and feasibility of the MPOA algorithm and provide references for further research and applications.

5.2. CEC2017 Comparative Experiment

CEC2017 is a testing suite used for evaluating and comparing algorithm performance. It provides 29 multi-feature functions, including unimodal, bimodal, and multimodal functions, along with corresponding evaluation metrics. This testing suite can be used for problems of different dimensions, including 2, 10, 30, 50, and 100 dimensions. To verify the acceptability of the improved algorithm MPOA's performance, we conducted experimental comparisons with traditional POA algorithm, Moth Flame Optimization algorithm MFO, and Grey Wolf Optimization algorithm GWO. In this section, we selected two sets of experiments for comparison, with dimension D as 30 and dimension D as 50. We compared the experimental data from five aspects: optimal value, mean square value, mean value, median, and worst value. The specific experimental data are shown in Tables 1-2.

Through the data analysis of Tables 1-2, this paper proposes that the MPOA algorithm demonstrates competitive performance in solving CEC2017 benchmark functions with dimensions of 30 and 50. Particularly, it shows significant advantages in functions F3, F10, F11, F13, F14, F15, and F18. These results indicate that the MPOA algorithm possesses high performance and adaptability in multi-dimensional problems. These findings are of great significance for further research and practical application of optimization algorithms in real-world problems.

5.3. Path Planning Simulation Experiment

In order to verify the effectiveness of the MPOA algorithm in mobile robot path planning, Matlab 2021a was used to simulate the program, and simulations were conducted in two grid map environments, M1 and M2, and compared with the traditional POA algorithm [15], traditional Sparrow Search Algorithm (SSA), and traditional Gray Wolf Optimization Algo-

Table 1
Experimental Comparative Data for Dimension D=30

F	Metrics	MFO	GWO	POA	MPOA
F1	min	1.7560E+09	4.4542E+08	1.1160E+10	1.5320E+08
	std	7.8575E+09	1.3470E+09	4.5683E+09	2.0933E+09
	avg	1.0940E+10	2.5543E+09	1.7488E+10	1.5353E+09
	median	9.7989E+09	2.4079E+09	1.7287E+10	8.5161E+08
	worse	3.3101E+10	5.9399E+09	3.0084E+10	9.4718E+09
F3	min	1.3244E+05	4.8276E+04	2.7936E+04	2.4066E+04
	std	5.4257E+04	1.7350E+04	8.8355E+03	6.1303E+03
	avg	2.0122E+05	7.5784E+04	4.8026E+04	4.0269E+04
	median	1.9614E+05	7.2861E+04	4.8848E+04	4.0862E+04
	worse	3.9200E+05	1.1778E+05	6.4789E+04	4.9677E+04
F4	min	5.4756E+02	5.2884E+02	7.2404E+02	5.4334E+02
	std	1.0442E+03	1.0254E+02	1.6142E+03	1.5740E+02
	avg	1.3945E+03	6.6378E+02	2.7961E+03	6.4821E+02
	median	9.8753E+02	6.2848E+02	2.4671E+03	6.0628E+02
	worse	4.7418E+03	9.3495E+02	7.0207E+03	1.4283E+03
F5	min	5.9865E+02	5.7585E+02	6.9587E+02	7.2533E+02
	std	5.7977E+01	5.7952E+01	3.8575E+01	2.5319E+01
	avg	7.1049E+02	6.3553E+02	7.7944E+02	7.7516E+02
	median	7.0103E+02	6.2240E+02	7.7990E+02	7.7613E+02
	worse	8.4984E+02	8.1898E+02	8.5142E+02	8.0534E+02
F6	min	6.2373E+02	6.0466E+02	6.4974E+02	6.4250E+02
	std	8.1070E+00	7.6552E+00	5.6680E+00	5.6455E+00
	avg	6.3652E+02	6.1511E+02	6.6311E+02	6.5867E+02
	median	6.3583E+02	6.1402E+02	6.6272E+02	6.5917E+02
	worse	6.5822E+02	6.4214E+02	6.7279E+02	6.6730E+02
F7	min	8.8931E+02	8.4573E+02	1.0939E+03	1.0120E+03
	std	1.5723E+02	5.2188E+01	6.0082E+01	9.7418E+01
	avg	1.0989E+03	9.1976E+02	1.2835E+03	1.2279E+03
	median	1.0380E+03	9.0906E+02	1.3021E+03	1.2672E+03
	worse	1.4001E+03	1.0043E+03	1.3615E+03	1.3592E+03
F8	min	9.3125E+02	8.5967E+02	9.5831E+02	9.4196E+02
	std	3.8595E+01	2.2200E+01	2.2135E+01	1.8385E+01
	avg	1.0040E+03	9.0587E+02	1.0073E+03	9.8206E+02
	median	1.0064E+03	9.0559E+02	1.0108E+03	9.8092E+02
	worse	1.0915E+03	9.5311E+02	1.0418E+03	1.0188E+03
F9	min	3.4690E+03	1.5890E+03	4.4531E+03	5.5195E+03
	std	2.3967E+03	1.3192E+03	6.3650E+02	2.2766E+02
	avg	7.6607E+03	3.2220E+03	6.1285E+03	5.9036E+03
	median	7.5305E+03	2.8978E+03	6.1794E+03	5.8511E+03
	worse	1.2604E+04	7.7224E+03	7.5284E+03	6.5714E+03
F10	min	4.4357E+03	3.9848E+03	4.4575E+03	4.9916E+03
	std	8.9990E+02	1.7875E+03	4.6762E+02	3.8522E+02
	avg	5.6193E+03	5.6983E+03	5.4683E+03	5.5060E+03
	median	5.4621E+03	5.1215E+03	5.4870E+03	5.4146E+03
	worse	7.8650E+03	9.7767E+03	6.2369E+03	6.6801E+03

F	Metrics	MFO	GWO	POA	MPOA
F11	min	1.3470E+03	1.3879E+03	1.3935E+03	1.2622E+03
	std	5.9146E+03	1.4661E+03	7.1075E+02	3.8761E+02
	avg	6.1782E+03	2.8744E+03	2.2187E+03	1.4431E+03
	median	3.9158E+03	2.3997E+03	2.0097E+03	1.3744E+03
	worse	3.0571E+04	7.6574E+03	4.8237E+03	3.4720E+03
F12	min	1.9572E+06	3.5371E+06	7.9285E+07	9.8764E+06
	std	2.0369E+08	1.1931E+08	1.5729E+09	6.7209E+08
	avg	1.3572E+08	1.1902E+08	1.5419E+09	2.3710E+08
	median	2.3357E+07	6.8571E+07	9.3520E+08	4.7187E+07
	worse	6.3892E+08	4.2281E+08	5.8658E+09	3.2754E+09
F13	min	1.5573E+04	1.1457E+05	1.3024E+05	6.2978E+04
	std	3.0722E+08	1.9183E+07	7.2336E+08	1.2295E+06
	avg	9.1156E+07	9.3256E+06	1.6815E+08	4.6887E+05
	median	3.6001E+05	3.8924E+05	1.2549E+06	1.6624E+05
	worse	1.3577E+09	6.4117E+07	3.9765E+09	6.7475E+06
F14	min	2.1277E+04	7.8602E+03	2.0694E+03	1.9479E+03
	std	2.0595E+06	8.5452E+05	5.9730E+04	2.7704E+04
	avg	1.1818E+06	7.1266E+05	4.3936E+04	2.7173E+04
	median	4.7305E+05	2.8271E+05	9.0742E+03	1.7064E+04
	worse	1.0503E+07	2.9621E+06	1.9245E+05	9.1712E+04
F15	min	2.8186E+03	1.8364E+04	1.2287E+04	1.0567E+04
	std	5.1537E+04	3.2484E+06	6.2290E+04	1.8046E+04
	avg	5.3092E+04	1.6589E+06	5.9741E+04	3.0710E+04
	median	3.9117E+04	1.2021E+05	4.2305E+04	2.2445E+04
	worse	2.1979E+05	1.1473E+07	3.1176E+05	7.9524E+04
F16	min	2.2734E+03	2.2052E+03	2.6484E+03	2.6259E+03
	std	3.4425E+02	4.1159E+02	3.7007E+02	2.7453E+02
	avg	3.0289E+03	2.7277E+03	3.2541E+03	3.0722E+03
	median	3.0289E+03	2.5943E+03	3.1916E+03	3.0796E+03
	worse	3.5968E+03	3.6483E+03	4.1431E+03	3.5946E+03
F17	min	1.9316E+03	1.8391E+03	1.8343E+03	1.8274E+03
	std	3.5786E+02	1.6858E+02	1.9900E+02	1.7830E+02
	avg	2.5256E+03	2.1257E+03	2.2884E+03	2.1679E+03
	median	2.5226E+03	2.1326E+03	2.3086E+03	2.1665E+03
	worse	3.2872E+03	2.4824E+03	2.7518E+03	2.5712E+03
F18	min	1.7614E+05	6.3944E+04	3.2999E+04	1.9368E+04
	std	7.5124E+06	5.6556E+06	9.8713E+05	2.6044E+05
	avg	6.0094E+06	3.9586E+06	4.7103E+05	2.6315E+05
	median	2.9738E+06	1.8134E+06	1.8078E+05	1.5457E+05
	worse	2.7219E+07	2.8099E+07	5.5206E+06	1.1602E+06
F19	min	5.5405E+03	2.3144E+04	8.5120E+04	1.6934E+04
	std	3.6680E+07	6.8542E+05	3.5842E+06	4.1573E+05
	avg	1.1925E+07	6.8487E+05	2.3634E+06	4.4139E+05
	median	1.1552E+05	5.0358E+05	1.5870E+06	3.1093E+05
	worse	1.7905E+08	3.3649E+06	2.0015E+07	1.7516E+06
F20	min	2.1832E+03	2.2249E+03	2.3376E+03	2.4693E+03
	std	2.7130E+02	1.9500E+02	1.0740E+02	1.9273E+02
	avg	2.7128E+03	2.5419E+03	2.5190E+03	2.8693E+03
	median	2.7231E+03	2.5278E+03	2.5181E+03	2.8875E+03
	worse	3.2404E+03	2.8670E+03	2.7251E+03	3.3419E+03

F	Metrics	MFO	GWO	POA	MPOA
F21	min	2.4293E+03	2.3700E+03	2.4563E+03	2.2650E+03
	std	4.3067E+01	2.8701E+01	5.1156E+01	7.7490E+01
	avg	2.4945E+03	2.4158E+03	2.5771E+03	2.4996E+03
	median	2.4932E+03	2.4052E+03	2.5711E+03	2.5113E+03
	worse	2.5635E+03	2.5044E+03	2.6997E+03	2.6269E+03
F22	min	2.6398E+03	2.4583E+03	3.3054E+03	2.4465E+03
	std	1.4103E+03	2.0306E+03	1.5503E+03	2.1100E+03
	avg	6.4934E+03	5.4138E+03	6.0834E+03	6.5699E+03
	median	6.8891E+03	5.9035E+03	6.2535E+03	7.5720E+03
	worse	8.8902E+03	1.0215E+04	8.2037E+03	8.4143E+03
F23	min	2.7554E+03	2.7360E+03	2.9475E+03	2.7829E+03
	std	4.1231E+01	4.9803E+01	7.2962E+01	7.0405E+01
	avg	2.8359E+03	2.8010E+03	3.0550E+03	2.8707E+03
	median	2.8311E+03	2.7895E+03	3.0547E+03	2.8645E+03
	worse	2.9233E+03	2.9934E+03	3.2292E+03	3.0965E+03
F24	min	2.9007E+03	2.9054E+03	3.0733E+03	2.8869E+03
	std	3.3477E+01	5.7654E+01	7.8054E+01	5.2922E+01
	avg	2.9850E+03	2.9740E+03	3.2321E+03	3.0211E+03
	median	2.9877E+03	2.9536E+03	3.2421E+03	3.0287E+03
	worse	3.0630E+03	3.1349E+03	3.4379E+03	3.0964E+03
F25	min	2.9336E+03	2.9263E+03	3.0610E+03	2.9401E+03
	std	3.7767E+02	6.9652E+01	2.4166E+02	3.9461E+01
	avg	3.3813E+03	3.0254E+03	3.3449E+03	3.0182E+03
	median	3.2600E+03	3.0142E+03	3.2780E+03	3.0204E+03
	worse	4.4852E+03	3.2844E+03	4.1235E+03	3.0873E+03
F26	min	3.8938E+03	4.2237E+03	4.0726E+03	3.4249E+03
	std	6.4277E+02	4.4006E+02	1.3633E+03	1.8152E+03
	avg	5.8484E+03	5.0238E+03	7.4506E+03	5.7696E+03
	median	5.8255E+03	4.8937E+03	7.6673E+03	6.0688E+03
	worse	7.9196E+03	6.2523E+03	9.2217E+03	8.4634E+03
F27	min	3.2094E+03	3.2314E+03	3.3012E+03	3.2302E+03
	std	2.3145E+01	2.5058E+01	7.9952E+01	2.9408E+01
	avg	3.2480E+03	3.2749E+03	3.4086E+03	3.2701E+03
	median	3.2449E+03	3.2723E+03	3.3865E+03	3.2634E+03
	worse	3.3141E+03	3.3370E+03	3.6204E+03	3.3392E+03
F28	min	3.3677E+03	3.3714E+03	3.4931E+03	3.3623E+03
	std	8.4850E+02	1.5852E+02	6.2741E+02	5.1939E+01
	avg	4.3913E+03	3.5465E+03	4.3227E+03	3.4433E+03
	median	4.1340E+03	3.4928E+03	4.2845E+03	3.4317E+03
	worse	6.2811E+03	3.9155E+03	6.1631E+03	3.5793E+03
F29	min	3.7485E+03	3.6799E+03	3.8150E+03	3.7805E+03
	std	3.6448E+02	1.9709E+02	4.3219E+02	3.1846E+02
	avg	4.2009E+03	4.0150E+03	4.7444E+03	4.2426E+03
	median	4.1239E+03	3.9881E+03	4.6889E+03	4.2453E+03
	worse	5.0875E+03	4.4674E+03	6.3031E+03	5.1866E+03
F30	min	1.6999E+04	1.8601E+06	2.1171E+06	3.8008E+05
	std	1.4751E+06	1.2102E+07	8.8374E+06	3.5472E+06
	avg	1.0034E+06	1.4539E+07	1.1364E+07	5.6332E+06
	median	2.9816E+05	9.9051E+06	8.1183E+06	5.0668E+06
	worse	5.6395E+06	4.1396E+07	4.3863E+07	1.6210E+07

Table 2
Experimental Comparative Data for Dimension D=50

F	Metrics	MFO	GWO	POA	MPOA
F1	min	1.4290E+10	4.0800E+09	3.4628E+10	3.0588E+09
	std	1.6337E+10	4.5068E+09	9.4800E+09	3.6856E+09
	avg	3.5817E+10	1.1525E+10	5.4322E+10	8.0595E+09
	median	3.3137E+10	1.1543E+10	5.2173E+10	7.4397E+09
	worse	8.2591E+10	2.2011E+10	7.2626E+10	1.8590E+10
F3	min	2.9859E+05	1.4263E+05	8.4292E+04	1.0835E+05
	std	8.0072E+04	4.2716E+04	2.3001E+04	1.2175E+04
	avg	4.4314E+05	2.2355E+05	1.3362E+05	1.3580E+05
	median	4.4543E+05	2.1938E+05	1.3279E+05	1.3594E+05
	worse	5.8704E+05	3.1228E+05	2.0155E+05	1.5866E+05
F4	min	1.2026E+03	8.1375E+02	5.4961E+03	8.8140E+02
	std	2.5451E+03	6.6452E+02	2.1501E+03	5.7993E+02
	avg	4.9416E+03	1.7692E+03	8.9501E+03	1.6469E+03
	median	4.4068E+03	1.6189E+03	8.9781E+03	1.5048E+03
	worse	1.2640E+04	3.4934E+03	1.3596E+04	3.4294E+03
F5	min	8.5008E+02	7.0694E+02	8.7615E+02	8.8969E+02
	std	7.6296E+01	3.5391E+01	4.0571E+01	2.0518E+01
	avg	1.0094E+03	7.8700E+02	9.4918E+02	9.2284E+02
	median	1.0175E+03	7.8135E+02	9.4853E+02	9.2613E+02
	worse	1.1529E+03	8.5034E+02	1.0204E+03	9.5292E+02
F6	min	6.3690E+02	6.1907E+02	6.6087E+02	6.5742E+02
	std	1.1599E+01	4.8933E+00	5.5844E+00	4.1767E+00
	avg	6.6543E+02	6.2850E+02	6.7274E+02	6.6870E+02
	median	6.6671E+02	6.3012E+02	6.7443E+02	6.6833E+02
	worse	6.8663E+02	6.3597E+02	6.8402E+02	6.7594E+02
F7	min	1.4384E+03	1.0473E+03	1.6411E+03	1.4991E+03
	std	3.3627E+02	7.3142E+01	7.8132E+01	8.3242E+01
	avg	2.0236E+03	1.1926E+03	1.8038E+03	1.7435E+03
	median	2.0092E+03	1.1742E+03	1.8090E+03	1.7737E+03
	worse	2.6319E+03	1.3544E+03	1.9493E+03	1.8343E+03
F8	min	1.1979E+03	9.8314E+02	1.2179E+03	1.1747E+03
	std	6.8202E+01	5.9055E+01	3.3269E+01	1.8995E+01
	avg	1.3216E+03	1.0834E+03	1.2773E+03	1.2130E+03
	median	1.3189E+03	1.0755E+03	1.2711E+03	1.2138E+03
	worse	1.4846E+03	1.3199E+03	1.3425E+03	1.2523E+03
F9	min	1.3464E+04	5.1102E+03	1.3854E+04	1.4550E+04
	std	3.8215E+03	4.9555E+03	2.3594E+03	2.6011E+03
	avg	2.2025E+04	1.2651E+04	1.9470E+04	1.9076E+04
	median	2.2656E+04	1.2386E+04	1.9313E+04	1.8887E+04
	worse	3.0890E+04	2.4863E+04	2.4915E+04	2.4566E+04
F10	min	7.1384E+03	7.1237E+03	8.0821E+03	8.6223E+03
	std	9.7700E+02	2.1899E+03	7.1657E+02	1.1042E+03
	avg	8.9592E+03	9.0480E+03	9.7342E+03	1.0550E+04
	median	8.9726E+03	8.2642E+03	9.6665E+03	1.0542E+04
	worse	1.1065E+04	1.5443E+04	1.1080E+04	1.2892E+04

F	Metrics	MFO	GWO	POA	MPOA
F11	min	3.3326E+03	4.3429E+03	3.2924E+03	1.9104E+03
	std	1.4128E+04	3.3617E+03	2.5328E+03	9.1565E+02
	avg	2.1464E+04	9.7222E+03	8.5252E+03	3.6262E+03
	median	1.6684E+04	9.1446E+03	8.8377E+03	3.7486E+03
	worse	6.5226E+04	1.7329E+04	1.3542E+04	5.0296E+03
F12	min	3.9083E+08	2.0616E+08	5.3903E+09	6.1706E+07
	std	5.2212E+09	2.2097E+09	1.0565E+10	2.5808E+09
	avg	5.7228E+09	2.3735E+09	2.0084E+10	1.5484E+09
	median	3.3998E+09	1.6041E+09	1.8894E+10	5.1878E+08
	worse	1.9446E+10	8.7327E+09	3.9790E+10	1.1336E+10
F13	min	1.0537E+05	5.2284E+06	2.0219E+08	5.5974E+06
	std	1.6020E+09	3.1267E+08	3.5388E+09	1.5791E+09
	avg	1.0196E+09	2.8543E+08	3.3077E+09	4.0686E+08
	median	2.2556E+07	2.2475E+08	1.5416E+09	9.7627E+07
	worse	6.2608E+09	1.5350E+09	1.1548E+10	8.7425E+09
F14	min	9.5380E+04	9.2459E+04	8.2062E+04	6.0104E+04
	std	2.6797E+06	2.3335E+06	5.3369E+05	3.3825E+05
	avg	2.2419E+06	2.3778E+06	7.0915E+05	5.2050E+05
	median	1.6124E+06	1.5645E+06	6.0468E+05	4.5859E+05
	worse	1.4158E+07	8.3721E+06	2.1907E+06	1.1712E+06
F15	min	9.8104E+03	1.0658E+05	4.0817E+05	3.5652E+04
	std	6.0057E+08	7.9762E+07	3.4831E+08	5.6777E+06
	avg	1.5842E+08	3.6683E+07	2.4723E+08	1.5784E+06
	median	1.1720E+05	1.7525E+07	6.0390E+07	3.1627E+05
	worse	2.8503E+09	4.3711E+08	1.2107E+09	3.1441E+07
F16	min	3.5025E+03	2.7586E+03	3.5692E+03	2.8834E+03
	std	5.5253E+02	6.2238E+02	6.7805E+02	3.4024E+02
	avg	4.4595E+03	3.6591E+03	4.6357E+03	3.5615E+03
	median	4.4432E+03	3.4520E+03	4.6963E+03	3.5668E+03
	worse	5.4713E+03	5.2201E+03	5.8728E+03	4.4363E+03
F17	min	3.3453E+03	2.7641E+03	2.5968E+03	2.5189E+03
	std	5.2125E+02	3.3114E+02	4.7194E+02	3.1654E+02
	avg	4.1427E+03	3.1787E+03	3.7079E+03	3.2871E+03
	median	4.1841E+03	3.1320E+03	3.6935E+03	3.1801E+03
	worse	5.4792E+03	4.1524E+03	4.5039E+03	4.0008E+03
F18	min	7.9204E+05	1.4299E+06	1.0242E+06	6.5958E+05
	std	1.9280E+07	1.6911E+07	7.5481E+06	4.4026E+06
	avg	1.3093E+07	1.6866E+07	6.8556E+06	2.8380E+06
	median	6.9459E+06	1.0063E+07	4.5713E+06	1.6771E+06
	worse	1.0298E+08	6.3072E+07	3.7837E+07	2.5217E+07
F19	min	1.8058E+04	1.0616E+05	3.5094E+05	4.9628E+04
	std	4.7356E+07	2.9381E+07	3.3146E+08	1.1290E+06
	avg	1.3992E+07	1.3887E+07	1.6453E+08	1.1048E+06
	median	6.7022E+05	2.6476E+06	1.5620E+07	6.6174E+05
	worse	1.9658E+08	1.3930E+08	1.3501E+09	5.4038E+06
F20	min	3.0367E+03	2.6532E+03	2.8345E+03	3.1234E+03
	std	3.1336E+02	4.3140E+02	1.8097E+02	2.2984E+02
	avg	3.5971E+03	3.3207E+03	3.1733E+03	3.6965E+03
	median	3.5972E+03	3.2321E+03	3.1530E+03	3.7305E+03
	worse	4.3498E+03	4.1547E+03	3.5812E+03	4.0162E+03

F	Metrics	MFO	GWO	POA	MPOA
F21	min	2.6137E+03	2.5002E+03	2.7468E+03	2.6461E+03
	std	8.2179E+01	4.3581E+01	7.3378E+01	6.8697E+01
	avg	2.7694E+03	2.5719E+03	2.8877E+03	2.7732E+03
	median	2.7690E+03	2.5679E+03	2.8763E+03	2.7617E+03
	worse	2.9436E+03	2.6599E+03	3.0389E+03	2.9737E+03
F22	min	9.3004E+03	8.6960E+03	9.7977E+03	1.1557E+04
	std	9.7078E+02	2.2100E+03	8.4355E+02	5.2564E+02
	avg	1.1141E+04	1.0823E+04	1.1912E+04	1.2695E+04
	median	1.1032E+04	1.0272E+04	1.1972E+04	1.2836E+04
	worse	1.3224E+04	1.7411E+04	1.3432E+04	1.3603E+04
F23	min	3.0324E+03	2.9325E+03	3.4153E+03	3.1030E+03
	std	7.7316E+01	8.2419E+01	1.6732E+02	6.1492E+01
	avg	3.2033E+03	3.0816E+03	3.6836E+03	3.2137E+03
	median	3.1924E+03	3.0780E+03	3.6897E+03	3.2136E+03
	worse	3.3972E+03	3.3410E+03	4.0233E+03	3.3870E+03
F24	min	3.1369E+03	3.1015E+03	3.4554E+03	3.2449E+03
	std	5.5067E+01	9.8117E+01	1.4576E+02	9.0179E+01
	avg	3.2596E+03	3.2316E+03	3.8110E+03	3.3786E+03
	median	3.2663E+03	3.2043E+03	3.8314E+03	3.3533E+03
	worse	3.3736E+03	3.5364E+03	4.0167E+03	3.6082E+03
F25	min	3.5481E+03	3.2040E+03	4.9419E+03	3.5512E+03
	std	2.7879E+03	6.0030E+02	1.5027E+03	3.4745E+02
	avg	6.3201E+03	3.9922E+03	7.2773E+03	4.0372E+03
	median	5.2588E+03	3.7916E+03	7.0025E+03	4.0342E+03
	worse	1.4253E+04	5.4570E+03	1.0681E+04	4.9947E+03
F26	min	7.2262E+03	5.6350E+03	9.5006E+03	7.4236E+03
	std	6.9455E+02	7.3001E+02	1.3385E+03	1.3951E+03
	avg	8.7629E+03	7.0355E+03	1.3285E+04	1.0339E+04
	median	8.8864E+03	7.1199E+03	1.3484E+04	1.0219E+04
	worse	9.9569E+03	8.8258E+03	1.5714E+04	1.3131E+04
F27	min	3.4475E+03	3.4988E+03	3.7528E+03	3.4403E+03
	std	1.0736E+02	1.4518E+02	2.9622E+02	1.7144E+02
	avg	3.6494E+03	3.7587E+03	4.2899E+03	3.6949E+03
	median	3.6418E+03	3.7218E+03	4.2508E+03	3.6707E+03
	worse	3.9222E+03	4.0924E+03	4.8582E+03	4.1918E+03
F28	min	5.3453E+03	3.8223E+03	6.0391E+03	4.0347E+03
	std	9.1052E+02	6.0678E+02	7.3120E+02	2.2286E+02
	avg	8.2755E+03	4.8205E+03	7.1097E+03	4.4929E+03
	median	8.5233E+03	4.7335E+03	7.0666E+03	4.4842E+03
	worse	9.6339E+03	6.5296E+03	9.0748E+03	4.9638E+03
F29	min	4.2564E+03	4.3101E+03	5.5523E+03	4.5912E+03
	std	6.9230E+02	4.7607E+02	9.5542E+02	5.8591E+02
	avg	5.4157E+03	5.0888E+03	7.4758E+03	5.5518E+03
	median	5.3203E+03	5.1369E+03	7.4031E+03	5.3896E+03
	worse	7.6277E+03	6.2149E+03	9.5163E+03	6.9618E+03
F30	min	3.2461E+06	9.5712E+07	1.2069E+08	6.1319E+07
	std	1.0682E+08	5.4832E+07	5.7650E+08	1.7736E+07
	avg	5.5534E+07	1.9728E+08	3.9531E+08	8.4684E+07
	median	1.5206E+07	1.8565E+08	2.5389E+08	8.2339E+07
	worse	4.3562E+08	3.5260E+08	3.2967E+09	1.2472E+08

rithm (GWO). M1 is a relatively simple environment model of 2020, and M2 is a more complex environment model of 4040. Both models have the robot path planning starting from the bottom left corner node and ending at the top right corner node. The population size for both is set to 30, and the maximum number of iterations is set to 200. Under the M1 environment, the convergence curve of various algorithms is shown in Figure 3, and the simulation results of a specific path planning experiment for various algorithms are shown in Figure 4. The simulation experiment results indicate that the improved MPOA algorithm outperforms other algorithms in terms of convergence speed and path length in relatively simple map environments.

Figure 3

Convergence Curve Graph of Path in M1 Map Environment

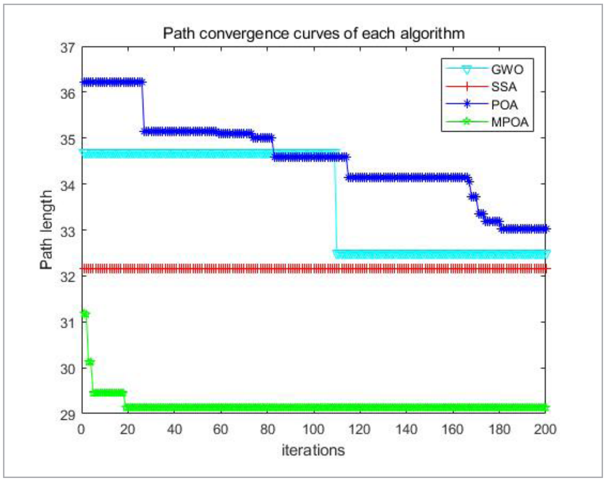
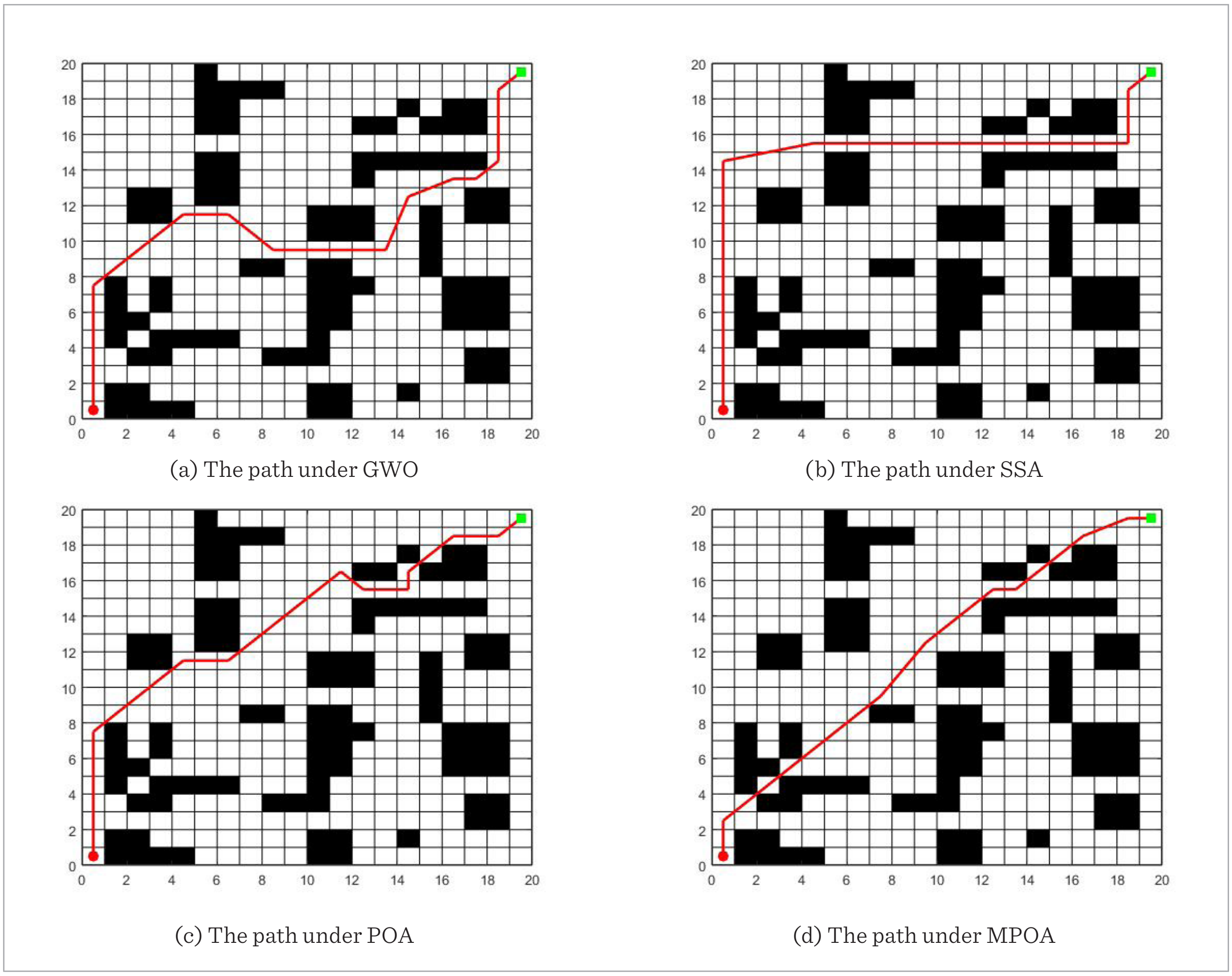


Figure 4

Route simulation results in M1 map environment



(a) The path under GWO

(b) The path under SSA

(c) The path under POA

(d) The path under MPOA

In the M2 environment, the convergence curves of various algorithm paths are shown in Figure 5, and the results of a single path planning simulation experiment for various algorithms are shown in Figure 6. The simulation experiment results show that the improved MPOA algorithm, in a relatively complex map environment, although converges slower than the SSA algorithm, it converges at 13 iterations, which is still relatively fast, and it has a better advantage in terms of path length.

To verify the stability of the algorithm, the algorithm was run 20 times in two different map environments, and the average path length, variance, and average path shortening rate were obtained as shown in Table 3. In the M1 map environment, the improved MPOA

Figure 5

Convergence Curve Graph of Path in M2 Map Environment

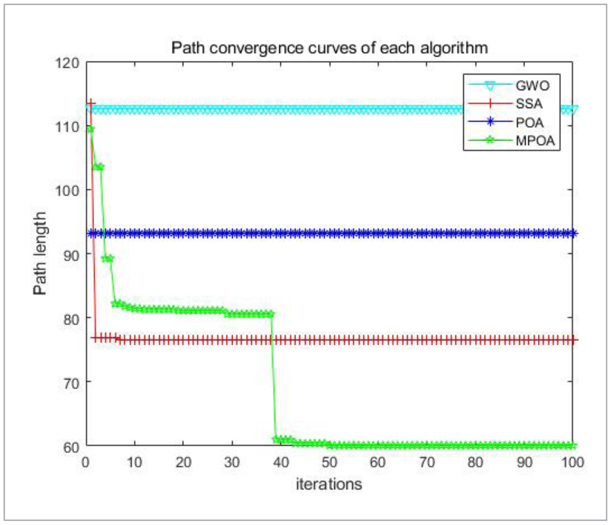


Figure 6

Route simulation results in M2 map environment

(a) The path under GWO

(b) The path under SSA

(c) The path under POA

(d) The path under MPOA

Table 3
Performance Comparison of 4 Algorithms Running 20 Times

Environment	M1			
algorithm	MPOA	POA	GWO	SSA
mean	29.10594	31.897415	31.39047	33.643415
std	1.014380	1.570274	2.865174	1.775402
MPOA average path shortening rate	/	0.08751414	0.10079715	0.16101263
Environment	M2			
algorithm	MPOA	POA	GWO	SSA
mean	67.06096	105.846095	115.235485	76.5132
std	6.580373	8.766548	10.233988	2.8421E-14
MPOA average path shortening rate	/	0.36642953	0.41805286	0.12353737

algorithm had an average path length of 29.10594 and a variance of 1.014380736, which was better than the other three algorithms. In the M2 map environment, the variance of the improved MPOA algorithm was 6.580373756, which was worse than the result of the SSA algorithm, but the average path length was 67.06096, which was better than the other three algorithms. This indicates that the proposed MPOA algorithm is feasible for mobile robot path planning in both simple and complex map environments, and it has good advantages in terms of convergence speed and shortest path.

6. Conclusions and Future Work

The POA algorithm is a heuristic optimization algorithm based on the behavior of pelicans, used to solve optimization problems. This algorithm simulates the foraging strategy of pelicans by iteratively searching the solution space to find the optimal solution. It features diversity, adaptability, and parallelism, and has been widely applied in fields such as function optimization, machine learning, and image processing. However, when the POA algorithm is applied to complex problems such as mobile robot path planning, it often suffers from issues of being trapped in local optima and premature convergence due to its population randomness and insufficient global exploration capability. To address these problems, this paper proposes a multi-strategy improved MPOA algorithm. The MPOA algorithm improves the random generation of the pelican population by using the Cubic chaotic map for initialization, enhancing the population's diversity. Additionally, an adaptive feedback regulation factor W is introduced to ensure that the updates of pelican positions occur within a certain range, solving the problem of blindly following local development in the later stage of the exploration phase. In the development stage of the algorithm, Levy flight strategy is first employed to update the positions of individual pelicans to maintain global optimization capability. Then, in the later stage of pelican position update, the Gaussian mutation mechanism is introduced to address the slow convergence issue in the Levy flight strategy.

To validate the performance of the improved algorithm, two different experiments are conducted in this paper. In the first experiment, the performance of the MPOA algorithm is compared with traditional POA algorithm, Moth Flame Optimization algorithm (MFO), and Grey Wolf Optimization algorithm (GWO) using 30-dimensional and 50-dimensional CEC 2017 benchmark functions. The results demonstrate the feasibility and advantages of the MPOA algorithm in solving various types of real optimization problems. In the second experiment, the proposed MPOA algorithm is simulated for mobile robot path planning in map environments of two different scales, 2020 and 4040, and compared with traditional POA algorithm, traditional Sparrow Search Algorithm (SSA), and traditional GWO algorithm. After 20 statistical data iterations, the results show that the MPOA algorithm exhibits superior path planning performance and algorithm stability in this experiment. In conclusion, the proposed MPOA algorithm addresses the drawbacks of the traditional POA algorithm in terms of slow convergence speed, low optimization accuracy, and poor robustness in engineering applications by improving the population initialization, position update strategy, and introducing Gaussian mutation mechanism. The experimental results demonstrate the feasibility and advantages of the MPOA algorithm in solving various types of real optimization problems and mobile robot path planning.

MPOA algorithm is an emerging optimization algorithm that improves upon the POA algorithm. The MPOA algorithm has the potential for broader application and development in the future. Researchers can continue to improve the algorithm's performance and efficiency to further enhance its applicability in practical problems. One method to improve the performance of the MPOA algorithm is optimizing parameter selection. By selecting parameters reasonably, the algorithm can better adapt to different types of problems. Researchers can determine the optimal parameter settings through experimentation and analysis to improve the algorithm's convergence speed and ability to solve complex problems. Another method to improve the MPOA algorithm is enhancing the neighborhood search strategy. Neighborhood search is an important step in optimization algorithms as it determines whether the algorithm can find better solutions. Researchers can design more effective neighborhood search strategies, such as introducing a combination of local and global search strategies, to improve the algorithm's search efficiency and problem-solving ability. Additionally, introducing new heuristic mechanisms is another approach to improving the MPOA algorithm. Heuristic mechanisms can help the algorithm explore the solution space better and guide the algorithm towards searching for better solutions. Researchers can design fitness functions and search strategies adapted to multi-objective problems to improve the performance of the MPOA algorithm in handling more complex practical problems. In conclusion, as an emerging optimization algorithm, the MPOA algorithm has the potential for broader application and development in the future. Researchers can further improve the algorithm's performance and efficiency by optimizing parameter selection, enhancing the neighborhood search strategy, and introducing new heuristic mechanisms to enable it to handle more complex practical problems.

Funding

This article was supported by the basic ability enhancement program for Young and Middle-Aged Teachers of Guangxi: 2022KY0767.

References

1. Al-Wesabi, F. N., Mengash, H. A., Marzouk, R., Alruwais, N., Allafi, R., Alabdan, R., Alharbi, M., Gupta, D. Pelican Optimization Algorithm with Federated Learning Driven Attack Detection Model in Internet of Things Environment. Future Generation Computer Systems, 2023,148, 118-127. https://doi.org/10.1016/j.future.2023.05.029
2. Chang, J., Ren, Y. Robot Path Planning Based on Improved Genetic Algorithm. Modular Machine Tool & Automatic Manufacturing Technique, 2023,02, 23-27. Doi:10.13462/J.Cnki.Mmtamt.2023.02.00
3. Gao, M., Zeng, X. Asynchronous Motor Fault Diagnosis based on Combination of VMD-ESA and IPOA-XGBOOST. Modern Electronics Technique.2024,47(02), 115-120. DOI:10.16652/j.issn.1004-373x.2024.02.022.
4. Hao, Z.-M., An, P.-J., Li, H.-Y., Zhao, T.-Y., Wang, L., Yang, C.-X. Mobile Robot Path Planning based on Enhanced Goal Heuristic Information Ant Colony Algorithm. Science Technology and Engineering, 2023,23(22), 9585-9591. DOI:10.12404/j.issn.1671-1815.2023.23.22.09585.
5. Huang, F., Jiang, S., Xu, J. The Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. Machinery Design & Manufacture, 12, Machinery Design & Manufacture, 2023(12), 194-198. DOI:10.19356/j.cnki.1001-3997.20230605.030.
6. Wang, G., Zhou, Q., Chen, H. Multi-Strategy Improved Sooty Tern Optimization Algorithm and its Engineering Application, 2023, (03), 28-34.DOI:10.19356/j.cnki.1001-3997.20221103.007.
7. Liu, B., Dong, X., Zhang, Z., Tu, C. Research on Robot Path Planning of Sparrow Search Particle Swarm Optimization Algorithm. Journal of North University of China (Natural Science Edition), 2023, 44(04), 374-380. DOI:10.3969/j.issn.1673-3193.2023.04.007.

8. Ma, X., Li, C., Tan, Y., Mei, H. Research of Artificial Fish Swarm Optimization Algorithm of Path Planning for Mobile Robot. Mechanical Science and Technology for Aerospace Engineering, 2024. https://doi.org/10.13433/j.cnki.1003-8728.20230103.

9. Mantegna, R. N. Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Processes. Physical Review E, 1994, 49(5), 4677-4683. https://doi.org/10.1103/PhysRevE.49.4677

10. Song, Y., Ma, P., Cheng, C. Mobile Robot Path Planning Based on Brainstorm Particle Swarm. Journal of Changchun University of Technology, 2023, 44(01), 38-44. DOI:10.15923/j.cnki.cn22-1382/t.2023.1.06.

11. Trojovský, P., Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. Sensors, 2022, 22(3), 855. https://doi.org/10.3390/s22030855

12. Wang, L., Wang, Y., Li, D., Wang, T. Research on Path Planning of Mobile Robot based on Improved Genetic Algorithm. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2024. https://doi.org/10.13245/j.hust.240403

13. Yang, G., Hu, H., Wang, X., Feng, S., Wang, F., Sun, J. Research on Image Matching Method Based on Mass Perturbation Pelican Optimization Algorithm. Journal of Zhengzhou University (Natural Science Edition), 2024. https://doi.org/10.13705/j.issn.1671-6841.2023047.

14. Zhang, C., Pei, Y.-H., Wang, X., Hou, H., Fu, L.-H. Symmetric Cross-Entropy Multi-Threshold Color Image Segmentation based on Improved Pelican Optimization Algorithm. PLOS ONE, 2023, 18(6), e0287573-e0287573. https://doi.org/10.1371/journal.pone.0287573

15. Zhang, J. Path Planning of Inspection Robot in Nuclear Environment Based on Pelican Optimization Algorithm. Machine Design & Research, 2023, 39(03), 27-30+34. DOI:10.13952/j.cnki.jofmdr.2023.0091.

16. Zhao, D., He, K., Zhao, Z. Path Planning for Mobile Robot Based on Improved PSO Algorithm. Transducer and Microsystem Technologies, 2023, 42(06), 150-153. DOI :10.13873/J.1000-9787(2023)06-0150-04.

17. Zhao, J.-T., Luo, X.-C., Liu, J.-M. Application of Improved Whale Optimization Algorithm in Robot Path Planning. Journal of Northeastern University (Natural Science), 2023, 44(08), 1065-1071. DOI:10.12068/j.issn.1005-3026.2023.08.001