Quantum-behaved Pigeon-inspired Optimization Algorithm based on Mutation Disturbance

Bingsong Xiao
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
58818252@qq.com

Rui Wang*
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
15353724115@163.com

Yongjian Yang
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
yangyongjian_king@126.com

Youwei Deng
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
dengyw_radar@163.com

Jundi Wang
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
qxwangjundi@sina.com

Wenjun Song
Aeronautics Engineering College
Air Force Engineering University
Xi'an, Shaanxi, China
412746417@qq.com

***Abstract*—Aiming at the shortage of the fine search ability in the Quantum-behaved Pigeon-inspired Optimization Algorithm, a new algorithm with mutation disturbance is proposed. Firstly, aiming at the map and compass operator characteristics of solve rough search, Cauchy mutation is adopted for the pigeons when they have not evolved at this stage, so as to enhance the global search ability. Then, aiming at the landmark operator characteristics of solve fine search, Gaussian mutation is adopted for the pigeons when they have not evolved at this stage, so as to enhance the local search ability. Finally, model of pigeons' number factor which likes map and compass factor is built to keep the pigeons in the landmark operator at a certain amount, so as to get the optimal solution. The function test results show that the search speed and accuracy of the improved algorithm are better than other popular algorithms.**

Keywords—swarm intelligence, pigeon-inspired optimization, Cauchy mutation, Gaussian mutation, number factor

I. INTRODUCTION

Optimization problems have a wide range of needs in modern science and technology. The use of swarm intelligence, an emerging evolutionary computing technology to solve optimization problems, has been paid more and more attention. In July 2017, the development plan of new generation for artificial intelligence by the State Council listed the key technologies of swarm intelligence as one of the eight key common technologies. In the theory of swarm intelligence, the key points are to break through the theory and method of organization, emergence and learning of swarm intelligence, and to establish an extensible and computable group intelligence excitation algorithm and model. The swarm intelligence algorithm realizes the purpose of optimization by simulating various swarm behaviors of social animals and utilizing information interaction and cooperation among individuals in the group[1], such as ant colony algorithm, particle swarm algorithm, hybrid frog leaping algorithm, artificial bee colony algorithm, firefly algorithm, pigeon-inspired optimization(PIO) algorithm. Among the many swarm intelligent algorithms, the PIO algorithm has received extensive attention since it was proposed by Duan in 2014[2]. The basic idea of PIO is derived from the independent homing behavior of the pigeons. The concept of PIO is simple and it is easy to implement. It has been successfully applied in various fields such as drone formation, control parameter optimization, image processing, medical imaging, and biological detection[3]. H. H. Li[4] proposed a BQPIO algorithm. Based on the convergence behavior of a single pigeon, the BQPIO algorithm is inspired by quantum mechanics to make the pigeons have quantum behavior. Its remarkable features are fewer control parameters, simple setup, strong search capability, and better global search capability. R. Hao[5] proposed a nonlinear change of the map and compass factor in the iterative process, and introduced the cross-concept of genetic algorithm in the pigeon group algorithm. H. B. Duan[6] introduced a predator escape mechanism to improve the overall performance of the pigeons in order to optimize the basic pigeon population. Aiming at the problem that PIO is easy to fall into local optimization, S. J. Zhang and H. B. Duan and others used the Gaussian mutation[7] and Cauchy mutation[8] to disturbance the pigeons, which effectively reduced the probability that the optimization result falls into local optimization.

Science and Technology on Avionics Integration Laboratory and Aviation Science Fund (20175596020).

These research have effectively promoted the development of PIO algorithm, and made PIO algorithm converge faster. However, the problem of easy premature convergence and insufficient search ability remains unresolved. In response to this shortage, this paper proposes a new algorithm called Mutation Disturbance PIO(MDPIO). On the basis of maintaining the strong search ability of the quantum PIO(QPIO) algorithm, the new algorithm uses different disturbance strategies according to the characteristics of its missions in different search stages. At the same time, in the evolution of the pigeon swarm, the number of pigeons is kept at a reasonable scale, so that it can enhance the local search ability as much as possible while maintaining rapid convergence.

II. PIO AND QPIO

A. PIO

The PIO algorithm uses two different operator models by simulating the different stages of the pigeon's search for the target, using different navigation tools. They are map and compass operator, landmark operator. In this model, virtual pigeons are used to simulate the navigation process.

1) Map and compass operator

The map and compass operator mimics the effects of the sun and the magnetic field on the pigeons, and directs the pigeons to their destinations. As the first stage of finding a target, a rough search is performed. In the early stages of this phase, the sun and magnetic fields have a greater impact on the pigeons. As the number of iterations increases, the effect gradually diminishes. According to the principle of the map and compass operator, the position and speed of the pigeon are firstly initialized. Then, during each iteration, the position and speed of the pigeons should be updated in the multi-dimensional search space. The position and speed of the i-th pigeon are recorded as $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $V_i = [v_{i1}, v_{i2}, \cdots, v_{iD}]$, where i = 1,2,…,N. N is the size of the swarm. Each pigeon updates its position X_i and speed V_i according to (1) and (2):

$$X_i^t = X_i^{t-1} + V_i^t \tag{1}$$

$$V_i^t = V_i^{t-1} \mathrm{e}^{-R\times t} + \mathrm{rand}(X_{\mathrm{gbest}} - X_i^{t-1}) \tag{2}$$

where R is the map and compass factor, which can reduce the flight speed of the pigeons as iteratively proceeds, and rand is a random number, rand∈[0,1]. N_c is the current number of iterations. X_{gbest} is the global optimal position for the current search and is the result of comparison with other pigeon positions. When the number of iterations reaches the preset number of iterations Nc1, the map and the compass operator is stopped, and the landmark operator is executed to continue execution.

The map and compass operator models can be represented in Fig. 1. The pigeon on the right is X_{gbest}. Other pigeons point to it with a red arrow. The pigeon can be seen as a compass, leading other pigeons to fly in a better direction. At the same time, each pigeon also has its own original flight direction. That is the direction of the black arrow. This direction is related to $V_i^{t-1} e^{-R\times t}$. The final flight direction of each pigeon is the vector sum of the two directions. That is the direction of the blue arrow.

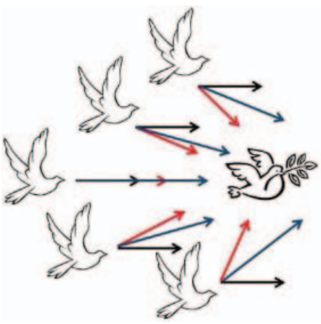


Fig. 1. Map and compass operator model

2) Landmark operator

The landmark operator is used to simulate the influence of landmarks on the pigeons in the navigation tool. When the pigeons fly close to their destination, they will rely more on nearby landmarks. If the pigeon is familiar with the landmark, it will fly directly to the destination. Otherwise, they will follow the pigeons that are familiar with the landmarks. As the second stage of searching the target, a fine search is performed.

In the landmark operator, half of the number of pigeons is decreased in each iteration. Because pigeons far from the destination are unfamiliar with the landmarks and they will no longer have the ability to resolve the path. Therefore, it should be abandoned. The quantity update method is as shown in (3).

$$N_p(t) = \frac{N_p(t-1)}{2} \tag{3}$$

X_{center} is the central location of the remaining pigeons and is used as a landmark, ie as a reference for flight. The update method is as shown in (4) and (5).

$$X_c(t) = \frac{\sum X_i(t) fitness(X_i(t))}{N_p \sum fitness(X_i(t)} \tag{4}$$

$$X_i(t) = X_i(t-1) + rand(X_c(t) - X_i(t-1)) \tag{5}$$

$X_c(t)$ is the center position of the pigeons for each iteration, and $fitness(X_i(t))$ is the fitness value of the pigeon. For solving the maximum problem, there is:

$$fitness(X_i(t)) = fitness(X_i(t)) \tag{6}$$

For solving the minimum problem, there is：

$$fitness(X_i(t)) = \frac{1}{fitness(X_i(t)) + \varepsilon} \tag{7}$$

After the N_{c2} iteration, the global optimal position P_g is obtained.

The landmark operator model can be represented by Fig. 2. The pigeon at the center of the right is the destination of each iteration. The pigeons can fly directly to the destination, such as the direction of the black arrow. However, the remaining pigeons far away from the destination follow other pigeons, such as the direction of the red arrow. This dynamic process effectively balances local and global searches.

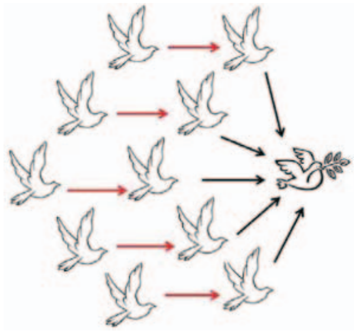


Fig. 2. Landmark operator model

B. QPIO

QPIO is a new global optimization algorithm based on PIO combined with the motion law of pigeons in one-dimensional δ potential well in quantum mechanics[4].

$m_{\mathrm{best}}(t)$ is the average of the pigeons at the t-th iteration, indicating the average best position in the population, defined as

$$m_{\mathrm{best}}(t+1) = \frac{1}{N_p} \sum_{i=1}^{N_p} P_i(t) \tag{8}$$

where $P_i(t)$ is the optimal position currently searched for each pigeon.

A new subgroup $P_i P_g(t+1)$ is created as

$$P_i P_g(t+1) = f(t+1) \times P_i(t) + (1 - f(t+1)) \times P_g(t) \tag{9}$$

where $P_g(t)$ is the optimal position currently searched for all the pigeons, and $f(t+1)$ is a random number with a uniform value in the range of 0~1.

Set $\omega(t)$ as the expansion-contraction factor, which is used to control the convergence speed of the algorithm. The value is the linear function of the number of iterations of the algorithm. That is

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{t}{t_{\max}} \tag{10}$$

where $t_{\max}$ is the total number of iterations, $\omega_{\max}$ and $\omega_{\min}$ are two positive constants.

The way to update individual pigeons is:

$$\begin{cases} X_i(t+1) = P_iP_g(t+1) + \omega(t+1) \times |m_{\text{best}}(t+1) - X_i(t)| \times \ln\frac{1}{u(t+1)}, \ f(t+1) \geq 0.5 \\ X_i(t+1) = P_iP_g(t+1) - \omega(t+1) \times |m_{\text{best}}(t+1) - X_i(t)| \times \ln\frac{1}{u(t+1)}, \ f(t+1) < 0.5 \end{cases} \quad (11)$$

where $u(t+1)$ is a random number with a uniform value in the range of 0~1.

The obvious difference between QPIO and PIO is that QPIO introduces a random logarithmic distribution of pigeon positions and proposes the concepts of m_{best} and P_iP_g. The logarithmic distribution of the pigeon position makes the pigeon's search space in each iteration as the entire real space, which can cover the entire feasible solution space and increase the ability to search the global optimal solution. The introduction of m_{best} has greatly improved the convergence performance of QPIO, because the cooperation ability between pigeons is stronger. A single pigeon cannot converge on by itself and must wait for other pigeons. Once an individual is far away from the swarm in the entire swarm, other individuals will wait for the outliers to keep up with the team, while during the waiting period, other individuals will do a large-scale global search near the global optimal g_{best} without quickly converge to g_{best}.

III. QPIO Algorithm based on Mutation Disturbance

A. Mutation Disturbance

Although the QPIO algorithm has disturbanced, the individual searches only in the vicinity of g_{best} in the global search. For the complex search space, the swarm is still easy to fall into the local optimum. That is to say, when the swarm gathers in a small range, the lack of swarm diversity is caused. So it is prone to local convergence in solving the multi-peak optimization problem or "stagnation" in a long period of time. The specific analysis is as follows.

Equation (11) can be rewritten as

$$X_i(t+1) = P_iP_g(t+1) \pm \omega(t+1) \times |m_{\text{best}}(t+1) - X_i(t)| \times \ln\frac{1}{u(t+1)} \quad (12)$$

Bringing the (9) into the (12),

$$X_i(t+1) = f(t+1)\left(P_i(t) - P_g(t)\right) + P_g(t) \pm \omega(t+1) \times |m_{\text{best}}(t+1) - X_i(t)| \times \ln\frac{1}{u(t+1)} \quad (13)$$

Equation (13) shows that in the search process, if the swarm moves into a small space, the current position of the pigeon X_i, the individual optimal position P_i of the pigeon and the global optimal position P_g are very close. This makes the value of the $(P_i(t) - P_g(t))$ in (13) very small, even zero. In this way, the effect of the item on the position update of the pigeon is small, which further leads to the lack of swarm diversity, which weakens the search ability of the swarm in a large range and falls into local optimum. Therefore, in order to maintain the diversity of the pigeon swarm in the whole search process, this paper adds mutation disturbance items to the QPIO's map and the compass operator and landmark operator to jump out of the local optimum.

In evolutionary computational theory, Gaussian mutation and Cauchy mutation are two commonly used disturbance techniques.

If the Cauchy mutation is used, then X is a random variable X=C that satisfies the Cauchy distribution, and its probability density function is as shown in (14):

$$f_C(x) = \frac{1}{\pi}\left(\frac{a}{a^2+x^2}\right) \quad (14)$$

where *a* is a scale parameter.

Its mutation value is:

$$a \times \tan[\pi(\text{rand} - 0.5)] \quad (15)$$

If Gaussian mutation is used, then X is a random variable $X=N(0,1)$ that satisfies the Gaussian distribution, and its probability density function is as shown in (16):

$$f_N(x) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right) \quad (16)$$

Its mutation value is:

$$a \times \tan[\pi(\text{randn} - 0.5)] \quad (17)$$

The biggest difference between the PIO algorithm and the previous swarm intelligence optimization algorithm is that two-stage search is adopted, and different operator models are adopted respectively. In the first stage, map and compass operator is mainly used to guide the pigeons to the destination, which is a rough search. In the second stage, landmark operator is mainly used to guide the pigeons to the final destination, which is a fine search.

For the different search requirements of the two stages, when selecting the perturbation algorithm, it needs to be considered separately. Cauchy mutation has advantages in jumping out of local optimum, while Gaussian mutation performs better in local convergence[9]. Therefore, for the PIO algorithm, the Cauchy mutation can be introduced into the map and compass operators, and the Gaussian mutation can be introduced into the landmark operator. In the first stage rough search falls into local optimum, the Cauchy mutation is used to increase the amplitude of the disturbance and expand the global search range. In the second stage refined search falls into local optimum, the Gaussian mutation is used to search in a small range, and the local search range is expanded to ensure convergence. This can avoid premature convergence and fall into the local optimal problem, and ensure that the landmark operator finds the global optimal.

B. Pigeon Swarm Evolution

The core idea of landmark operator is to enhance the local optimization ability of the pigeons and quickly obtain the optimal solution. The premise is that the pigeons quickly move closer to the optimal solution while maintaining a certain number of populations. However, in the landmark operator of the standard PIO, the number of pigeons will be reduced by half after each iteration, that is exponentially decreasing. The loss of the pigeons population is too large. Regardless of the initial number of pigeons, they will quickly drop to 2 or 1 (the result of rounding up is 2, the result of rounding is 1), and it remains 2 or 1 unchanged. So, the diversity of the algorithm is lost, which greatly affects the optimization performance of the algorithm. Therefore, this article uses pigeons number factor similar to the map and compass factor to adjust the number of pigeons in the landmark operator, as shown in (18).

$$N_p(t) = \text{PigeonNumInit} \times e^{-R\times t} \quad (18)$$

where PigeonNumInit is the initial number of pigeons.

For example, the initial number of pigeons is PigeonNumInit=50, and the number factor R is 0.1, 0.2, and 0.3. The relationship between the population number and the number of iterations is shown in Fig. 3.

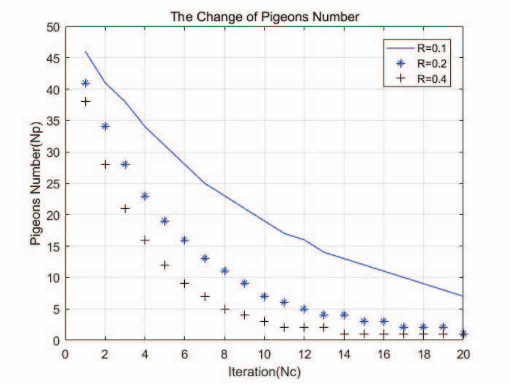


Fig. 3. Evolution of Pigeons Number

At the beginning of the landmark operator, a large number of pigeons are far away from the destination. At this time, the optimal contribution to the swarm is small, and it needs a lot of abandonment. Therefore, the number of pigeons is reduced at a faster rate. In the later stage of the landmark operator, the remaining pigeons gradually gather near the destination, and it is necessary to increase the influence on the central position while maintaining a certain population size. Therefore, the number of pigeons is reduced at a slower rate. This also facilitates the convergence of the algorithm.

C. Algorithm Flow

The flow of the MDPIO algorithm proposed in this paper is shown in Fig. 4.

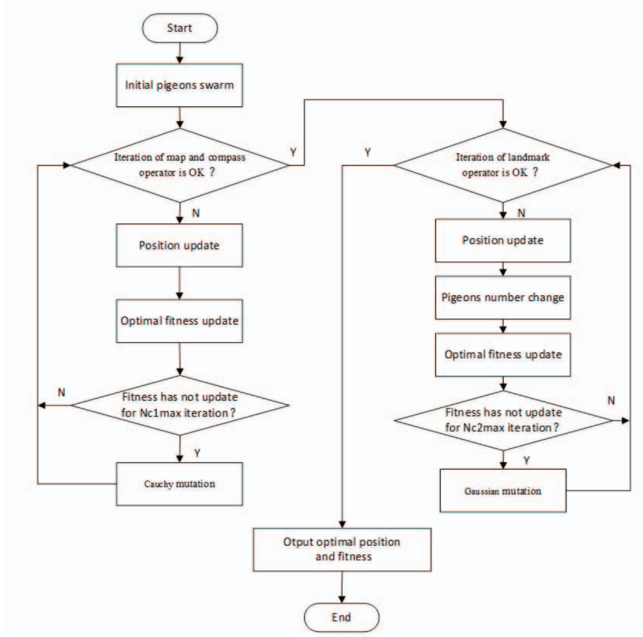


Fig. 4. Algorithm Flow of MDPIO

The computational overhead of various types of intelligent optimization algorithms is mainly due to the complexity of the objective function itself and the number of evaluations [10]. As can be seen from Fig. 4, the improved algorithm does not significantly increase the amount of computation for the objective function.

IV. NUMERICAL EXPERIMENT

A. Benchmark Functions And Other Algorithms For Comparison

In order to test the performance of the algorithm, this paper uses the standard test functions as in [4]: Shubert function, Rosenbrock function, Rastrigin function and Schaffer function. These functions include unimodal function, multimodal function, optimal value at (0, 0, ..., 0), an optimum value at other points, etc., as shown in Table Ⅰ.

TABLE I. TEST FUNCTION BASIC INFORMATION

Function Name	Function Expression	Solution Space	Global Minimum	Minimum Point
f1-Shubert	min f(x,y)={ $\sum_{i=1}^{5} icos[(i+1)x+i]\} \times \{\sum_{i=1}^{5} icos[(i+1)y+i]\}$	[-512,512]^2	-186.73	(-1.4251,0.8003)
f2- Rosenbrock	min f(x_i)={$\sum_{i=1}^{D-1}[100(x_i^2-x_{i+1})^2+(x_i-1)^2]$	[-2.048,20.48]	0	(1,1,…,1)
f3-Rastrigin	min f(x_i)= $\sum_{i=1}^{D-1}[x_i^2-10\cos(2\pi x_i)+10]$	[-5.12,5.12]	0	(0,0, … ,0)
f4-Schaffer	min f(x_1,x_2)=0.5+ $\frac{(\sin\sqrt{x_1^2+x_2^2})^2-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	[-10,10]	0	(0,0)

For QPIO, CMPIO, MDPIO, this paper sets the initial population of pigeons PigeonNumInit=50, map and compass factor, pigeon number factor R=0.2, map and compass operator iteration number N_{c1}=50, landmark operator iteration number N_{c2}=10, The Cauchy mutation condition N_{c1max}=3, the Gaussian mutation condition N_{c2max}=2, the map and compass operator mutation threshold Th_1=0.1, and the landmark operator mutation threshold Th_2=0.001.

B. Experimental Results And Analysis

The Monte Carlo simulations were performed 100 times for each of the four test functions. The statistical results of each algorithm for different function optimization are shown in Table II~V.

TABLE II. STATISTICAL RESULTS OF EACH ALGORITHM IN THE OPTIMIZATION OF SHUBERT FUNCTION

Algorithm	Optimum Value	Worst Value	Mean Value	Standard Deviation
QPIO	-1.867308e+02	-1.795573e+02	-1.864226e+02	8.082380e-01
CMPIO	-1.867309e+02	-1.851608e+02	-1.866043e+02	2.680723e-01
MDPIO	-1.867309e+02	-1.863996e+02	-1.867029e+02	5.216136e-02

TABLE III. STATISTICAL RESULTS OF EACH ALGORITHM IN THE OPTIMIZATION OF ROSENBROCK FUNCTION

Dimension	Algorithm	Optimum Value	Worst Value	Mean Value	Standard Deviation
	QPIO	1.536770e-13	6.327500e-03	2.975508e-04	1.089167e-03
10	CMPIO	3.532382e-09	5.312296e-04	2.725337e-05	6.098332e-04
	MDPIO	2.429259e-07	9.141363e-04	6.229231e-06	1.368541e-05
	QPIO	4.927919e-12	2.364817e-02	6.098172e-04	2.937569e-03
20	CMPIO	2.875068e-08	6.342611e-03	3.404107e-04	9.535897e-03
	MDPIO	1.456715e-08	8.768078e-03	7.007093e-05	1.166960e-04
	QPIO	7.384892e-14	3.043208e-02	6.609397e-04	3.429567e-03
30	CMPIO	7.534186e-07	4.263443e-03	4.573720e-04	8.435884e-03
	MDPIO	1.501853e-07	1.481173e-03	8.156573e-05	2.102469e-04

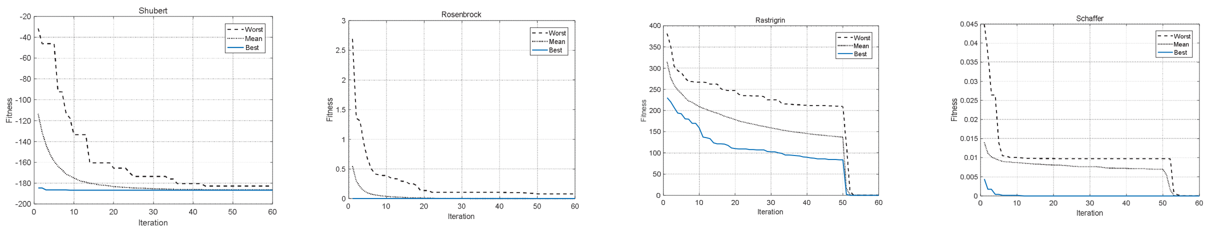
TABLE IV. STATISTICAL RESULTS OF EACH ALGORITHM IN THE OPTIMIZATION OF RASTRIGIN FUNCTION

Dimension	Algorithm	Optimum Value	Worst Value	Mean Value	Standard Deviation
	QPIO	0.000000e+00	3.656290e-08	1.080452e-08	4.364021e-08
10	CMPIO	3.599036e-10	9.158256e-08	1.596909e-09	1.838259e-09
	MDPIO	0.000000e+00	2.717155e-09	1.622737e-10	4.115657e-10
	QPIO	9.947598e-14	1.809783e-08	1.745212e-08	4.056805e-08
20	CMPIO	1.159227e-12	3.270251e-07	3.619590e-08	5.235973e-07
	MDPIO	0.000000e+00	3.137901e-08	3.948383e-10	3.144980e-09
	QPIO	1.048051e-13	1.287680e-07	2.712775e-09	1.329087e-08
30	CMPIO	1.729265e-12	5.977787e-07	4.668195e-08	7.632764e-07
	MDPIO	0.000000e+00	1.227558e-08	4.431321e-10	1.796138e-09

TABLE V. STATISTICAL RESULTS OF EACH ALGORITHM IN THE OPTIMIZATION OF SCHAFFER FUNCTION

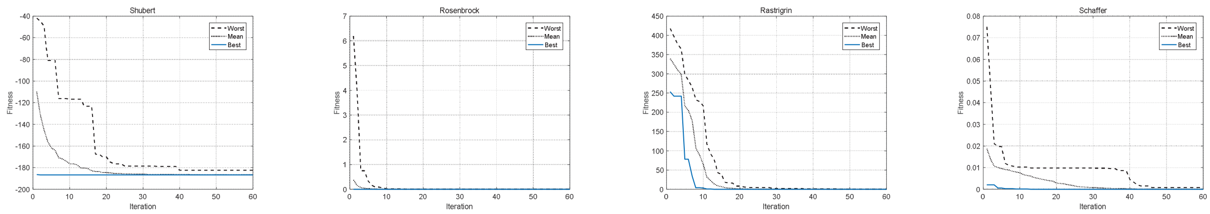
Algorithm	Optimum Value	Worst Value	Mean Value	Standard Deviation
QPIO	0.000000e+00	1.558070e-03	3.223010e-06	1.636524e-05
CMPIO	4.216066e-14	1.726083e-04	4.305834e-11	2.111004e-10
MDPIO	0.000000e+00	1.304490e-11	6.210482e-13	1.886796e-12

The average evolution curve of each function's optimal value, mean value and worst value with the number of iterations is shown in Fig. 5~7 for each function in 100 Monte Carlo simulations.



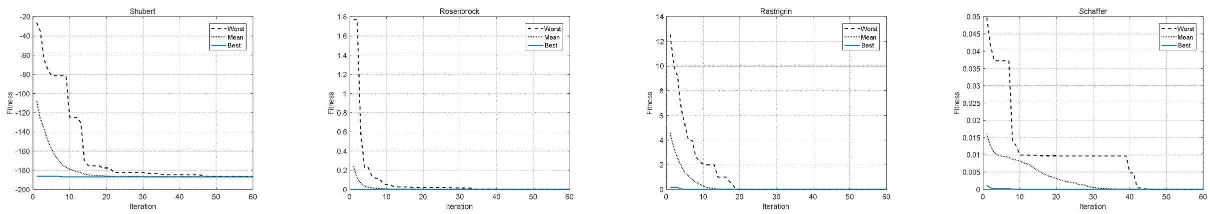
(a)Shubert (b)Rosenbrock (D=30) (c)Rastrigin (D=30) (d)Schaffer

Fig. 5. Fitness Average Evolution Process of QPIO Algorithm



(a)Shubert (b)Rosenbrock (*D*=30) (c)Rastrigin (*D*=30) (d)Schaffer

Fig. 6. Fitness Average Evolution Process of CMPIO Algorithm



(a)Shubert (b)Rosenbrock (*D*=30) (c)Rastrigin (*D*=30) (d)Schaffer

Fig. 7. Fitness Average Evolution Process of MDPIO Algorithm

According to the experimental results, it can be found that although three PIO algorithms tested can obtain the optimal solution, the number of iterations and the refined search ability required by each algorithm in obtaining the optimal solution are significantly different. The MDPIO algorithm adds a Cauchy disturbance to the map and the compass operator, and the speed of searching for the value near the optimal solution is significantly faster. Due to the addition of Gaussian disturbance to the landmark operator, the pigeons are able to perform a detailed search around the optimal value, while using the number of pigeons to maintain a certain size of the population, so that the optimal value is better than the other two algorithms.

V. CONCLUSION

In this paper, a quantum-behaved pigeon-inspired optimization algorithm with mutation disturbance is proposed. The difference between this algorithm and the traditional PIO algorithm is as follows. (1)For different mission characteristics of map and compass operator and landmark operator, different mutation disturbance methods are adopted respectively, so that the pigeons can quickly and roughly search for the vicinity of the global optimal value and then perform a detailed search. (2)By using the number factor of pigeons similar to the map and compass factor, the number of pigeons in the landmark operator is adjusted to ensure the diversity of the pigeons in the fine search, so that the optimal value can be searched.

In the next step, the PIO algorithm will be further studied in terms of the boundary processing and the processing of the center position of the remaining pigeons when the pigeon position is updated, and the search speed will be accelerated. At the same time, the optimization problem under discrete conditions will be studied and applied to solve practical problems.

ACKNOWLEDGMENT

This work was supported by Air Force Engineering University President Fund (XZJK2018003).

REFERENCES

[1] D. F. Wang, L. Meng, W. J. Zhao, Improved Bare Bones Particle Swarm Optimization with Adaptive Search Center, *Chinese Journal of Computers*,39(12):2652-2667,2016.

[2] H. B. Duan, P. X. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, *International Journal of Intelligent Computing and Cybernetics*, 7(1): 24-37, 2014.

[3] H. B. Duan, F. Ye, Research progress on pigeon-inspired optimization algorithm, *Journal of Beijing University of Technology*,43(1):1-7, 2017.

[4] H. H. Li, H. B. Duan, Bloch Quantum-behaved Pigeon-Inspired Optimization for Continuous Optimization Problems, Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference, Yantai, 2014: 2634-2638.

[5] R. Hao, D. Luo, H. B. Duan, Multiple UAVs mission assignment based on modified Pigeon-inspired optimization algorithm, Gguidance, Navigation and Control Conference. Yantai, China:IEEE,2015:2692-2697.

[6] H. B. Duan, H. X. Qiu, Y. M. Fan, Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization, *Sci Sin Tech*, 45(6):559-572, 2015.

[7] S. J. Zhang, H. B. Duan, Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration, *Chinese Journal of Aeronautics*, 28(1):200–205, 2015.

[8] H. B. Duan, Z. Y. Yang, Large civil aircraft receding horizon control based on Cauthy mutation pigeon inspired optimization, *Sci Sin Tech*, 48(3):277-288, 2018.

[9] K. T. Lan, C. H. Lan, Notes on the Distinction of Gaussian and Cauchy Mutations, *Intelligent Systems Design and Applications*, 9(11):272-277, 2008.

[10] W. Fang, J. Sun, W. B. Xu, Improved Quantum-behaved Particle Swarm Optimization Algorithm Based on Differential Evolution Operator and Its Application, *Journal of System Simulation*, 20(24):6740-6744, 2008.