A multi strategy improved pigeon-inspired optimization algorithm

Heng Liao, and *Huadong* Huang*

Guangxi International Business Vocational College, Nanning 530006, China

Abstract. Pigeon-inspired optimization algorithm is easy to fall into local optimization and low convergence accuracy in solving nonlinear optimization problems. In this paper, an improved pigeon-inspired optimization algorithm called Gaussian mixture pigeon-inspired optimization algorithm (GPIO) is proposed. In GPIO, the cubic mapping of chaotic mapping method is used to initialize the pigeon population, which increases the diversity of the population. Gaussian mutation operator is introduced to change the shortage that pigeon swarm algorithm is easy to fall into local optimization, and improve the convergence efficiency of the algorithm. The experimental results of 19 benchmark functions show that the algorithm has better optimization ability than other swarm intelligence algorithms.

Keywords: Pigeon-inspired optimization algorithm, chaotic mapping method, Gaussian mutation operator, Benchmark functions, Swarm intelligence algorithm.

1 Introduction

With the continuous improvement of science and technology, multi-objective optimization problem has become one of the main problems to be solved in engineering application. Multi optimization problem refers to a set of optimal solutions composed of the solutions of multiple sub-problems. In order to solve such problems, scholars have proposed some new meta heuristic algorithms [1], such as Particle Swarm Optimization (PSO) [2], Bat Algorithm (BA) [3], Butterfly Optimization Algorithm (BOA) [4], Flower Pollination Algorithm(FPA)[5], Teaching-learning-based Optimization(TLBO) [9], Crow search algorithm (CSA) [6], Whale Optimization Algorithm(WOA)[7], Gray Wolf Optimizer(GWO) [8], etc. These meta heuristic algorithms have good search ability and can solve most complex multi-objective optimization problems.

Pigeon-inspired optimization (PIO) is a new swarm intelligence optimization algorithm proposed by Duan [10] according to the homing behavior of pigeons in 2014. PIO, similar to other meta heuristic algorithms, has certain advantages in solving multi-objective

* Corresponding author: 2020210854001192@stu.gxun.edu.cn

optimization problems. However, PIO is easy to fall into local optimization and lack of global search ability. At present, many scholars have studied the defects of PIO and put forward many improvement strategies [11]. For example, in reference [12], the author proposed a cross pigeon swarm algorithm (IPIO) with cognitive factors. In the pigeon swarm algorithm, the map and compass operators are used to update the population position, and the nonlinear decreasing cognitive factors are added to optimize the global search ability of the algorithm; The compression factor is introduced into the landmark operator to improve the convergence accuracy of the algorithm. Through simulation, it is found that the operation time of the improved algorithm is relatively long and the operation speed is slow. In reference [13], the author proposed an improved pigeon swarm algorithm (PPIO) based on solving high-dimensional complex functions. The algorithm combines pigeon swarm algorithm with Powell optimization algorithm. The main idea is to use Powell optimization algorithm to optimize the optimization results of pigeon swarm algorithm and obtain better results. After improvement, the convergence accuracy and convergence speed of pigeon swarm algorithm are improved. However, this article does not point out the optimization effect of PPIO on low dimensional functions. In reference [14], the author proposed an improved pigeon swarm optimization algorithm for the path planning of robot movement. The pigeon swarm optimization algorithm is used to find the current best position, which improves the efficiency of path planning. In reference [15], the author solves the problem of low output efficiency and easy to fall into local optimization of the maximum power point tracking control method of photovoltaic system in multi peak state by improving the pigeon group optimization algorithm, which effectively improves the output The above improved algorithms all put forward their own improvement strategies for the improvement of PIO, and apply PIO to different engineering fields, but there are still some deficiencies. Therefore, it is of great significance to improve efficiency of PIO.

In this paper, a pigeon-inspired optimization algorithm with chaotic initialization and Gaussian mutation operator (GPIO) is proposed, which improves the global search ability and convergence accuracy of PIO algorithm. The improvement work mainly includes:

(1) population initialization uses cubic chaotic mapping method to enhance population diversity. (2) Gaussian mutation operator is introduced to make the algorithm jump out of the local optimization effectively and increase the global search ability.

(3) The parameter configuration of pigeon-inspired optimization algorithm is improved to optimize the convergence accuracy of the algorithm. Finally, the improved pigeon swarm algorithm is used to solve the optimal value of multi-objective optimization function and the minimum cost of power system economic dispatching, and the experimental results are given.

The structure of this paper is as follows. In section 2, the basic pigeon-inspired optimization algorithm will be briefly introduced. In section 3, the improved strategy will be described to produce an improved pigeon-inspired optimization. In section 4, the proposed improved algorithm is tested with 19 benchmark functions and compared with 6 algorithms. Finally, the conclusion is to given in Section 5.

2 Pigeon-inspired optimization algorithm

The pigeon-inspired optimization algorithm uses different navigation tools in different stages of the optimization process. Guilford [16] proposed that pigeons use magnetic field and compass as navigation tools to adjust the flight direction when they are far away from the target during flight. Braithwaite [17] found through experiments that the terrain will affect the homing behavior of pigeons during flight. Duan [10] proposed pigeon-inspired optimization algorithm according to the flight mechanism of pigeon swarm. The specific algorithm design is as follows:

When the pigeon group uses the map compass operator to navigate, it is first necessary to initialize the speed and position of each individual in the pigeon group. With the increase of the number of iterations, the position and speed of each individual in the pigeon group are updated according to the map and compass operator. Let the number of pigeons in the current flock be No and the dimension be dim, that is, the initialization speed v and position x of the flock are as follows:

$$\begin{aligned} x_i &= [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{i\dim}] \\ v_i &= [v_{i1}, v_{i2}, v_{i3}, \ldots, v_{i\dim}] \quad i = 1,2,3,\cdots,No. \end{aligned} \tag{1}$$

The pigeon group is updated according to the map and compass operator, and the formula is:

$$v_i^t = v_i^{t-1} \times e^{-Rt} + rand \times (x_{best} - x_i^{t-1}) \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

where, t represents the current iteration times, R represents the map and compass factor, and the value range is (0,1). $rand$ represents a random number with a value range from 0 to 1. x_{bext} represents the position of the global optimal solution found by the population before the t iteration process. When the number of iterations reaches the maximum number of iterations using map and compass operators, the pigeon group changes from using map and compass operators to using landmark operators. After the pigeon group uses equation (2) for multiple iterations to update the speed and position, it enters the landmark operator navigation stage. At this stage, pigeons with poor position in the population need to be abandoned in each iteration, and the population number becomes half of the number in the last iteration. The center position of the remaining pigeons is used as the forward direction. The specific expression is as follows:

$$x_c^{t-1} = \frac{\sum_{i=1}^{No^{t-1}} x_i^{t-1} f(x_i^{t-1})}{No^{t-1} \sum_{i=1}^{No^{t-1}} f(x_i^{t-1})} \tag{4}$$

$$No^t = \frac{No^{t-1}}{2} \tag{5}$$

$$x_i^t = x_i^{t-1} + rand \times (x_c^{t-1} - x_i^{t-1}) \tag{6}$$

where, x_c represents the central position of the pigeon group, and f is a function of obtaining the fitness value. According to different solving directions, it can be divided into the following two expressions:

$$f(x_i^{t-1}) = \begin{cases} \dfrac{1}{f(x_i^{t-1}) + \varepsilon} & \text{When finding the minimum problem} \\ f(x_i^{t-1}) & \text{When finding the maximum problem} \end{cases} \tag{7}$$

where, ε represents the infinitesimal value. When the number of iterations t reaches the critical value, the algorithm ends.

3 Gaussian mixture improved pigeon swarm algorithm (GPIO)

3.1 Chaos initialization

In the pigeon-inspired optimization algorithm, the pigeon position is generated by random initialization. The initial test position of the pigeon swarm is unevenly distributed in the solution space, which will affect the solution efficiency of the algorithm. Cubic mapping is a better method among many chaotic mapping sequence algorithms. In this paper, the cubic mapping method is used to preliminarily test the pigeon group position. The expression of cubic mapping method is as follows:

$$\begin{cases} y(i+1) = 4y(i)^3 - 3y(i), \\ -1 \le y(i) \le 1 \quad i = 0,1,2\cdots No \end{cases} \tag{8}$$

$$x_i = Ld + (1+y_i) \times \frac{Ud - Ld}{2} \tag{9}$$

where, Ld is the lower bound of the solution space and Ud is the upper bound of the solution space. Use equation (8) to generate the pigeon group, and then map the pigeon group position to the solution space according to equation (9), and initialize the pigeon group position.

3.2 Gaussian mutation strategy

Through experiments, it is found that the pigeon swarm algorithm is easy to fall into local optimization when updating the population position using map and compass operators. This paper introduces the idea of Gaussian mutation, randomly selects the pigeon position and its dimension for Gaussian mutation $Gaussian(\mu, \sigma^2)$, so that the pigeon swarm algorithm jumps out of the local optimization in the optimization process of this stage. The idea is divided into three steps:

Step 1: set the variation judgment factor gt, where the value of gt increases linearly to judge whether there is variation in this iteration. The expression of gt is as follows:

$$gt = (t / T_1) + 0.5 \tag{10}$$

where, t is the current number of iterations, and T_1 is the total number of iterations for pigeon group navigation using map and compass operators. The value of 0.5 in the above formula is a better parameter obtained through many experiments to control the number of population variation.

Step 2: Use the random number of $rand$ to judge whether it is greater than gt. if the value of gt is small, perform Gaussian mutation on the iterative population and execute step 3, otherwise execute the next iteration.

Step 3: Randomly select the position and dimension of the mutant pigeon, and the specific expression is as follows:

$$x_{(n,g)} = x_{(n,g)} \times Gaussian(\mu, \sigma^2) \tag{11}$$

In this paper, Gaussian mutation mechanism is used to update the position of pigeon group, which is conducive to jump out of local optimization in the process of population optimization and improve the optimization ability of the algorithm.

3.3 Decline factor

In the original pigeon swarm algorithm, pigeons with poor fitness value are discarded in each iteration, and the number of pigeons discarded is half of the population in the last iteration. The large number of abandoned pigeons will greatly reduce the diversity of the population in the later stage of iteration, affect the optimization ability of the algorithm and reduce the solution accuracy of the algorithm. Therefore, the decreasing factor w is introduced to affect the number of pigeons discarded in each iteration.

The specific expression is as follows:

$$w = 0.5 \times e^{-\frac{t}{T_2}} \tag{12}$$

$$No^t = No^{t-1} - No^{t-1} \times w \tag{13}$$

where T_2 is the total number of iterations using landmark operator.

GPIO algorithm increases the diversity of population through cubic mapping in chaotic mapping, jumps out of local optimization by using Gaussian mutation mechanism, and introduces decreasing factor to improve the solution accuracy of the algorithm. Therefore, GPIO algorithm can better solve the problems that PIO algorithm is easy to fall into local optimal solution and low algorithm accuracy.

4 Experimental simulation and result analysis

In order to verify the advanced and effective GPIO algorithm, 19 benchmark functions with different characteristics are selected to verify the performance of GPIO algorithm. These benchmark functions are divided into two categories: high-dimensional single-mode functions and high-dimensional multi-mode function. The expressions of the 19 benchmark functions are shown in Table 1:

In this paper, the traditional pigeon swarm algorithm (PIO), improved pigeon swarm algorithm (IPIO) [18], fruit fly optimization algorithm (FOA) [19], particle swarm optimization algorithm (PSO) [10] are compared with GPIO algorithm. Because the population size corresponding to the best effect of each algorithm in each study is inconsistent, in order to ensure the fairness of the experiment, the evaluation times are used as the termination condition of the algorithm, and all parameters of each algorithm are set according to the parameters proposed in the corresponding literature (including population size and key parameters). For the sake of fairness, we used a fixed population size for all algorithms: n = 50 individuals, all algorithms are executed in 30 independent runs. All algorithms are implemented in Matlab (version R2020a) and executed on HP computer (Windows 10,Inter Core i5-6300HQ, 2.3 GHz, 8GB RAM). Table 2 shows the data obtained by GPIO algorithm running independently with PIO, IPIO, FOA and PSO for 30 times, where best is the optimal value obtained by running, worst is the worst value obtained by running for 30 times, mean represents the average value of the results of 30 times, STD represents the variance, and timec represents the average time of each run.

Table 1. Benchmark test function.

Benchmark function	Range	optimal value
$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$	$x_i \in [-100,100]$	0
$f_2(\vec{x}) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$	$x_i \in [-100,100]$	0
$f_3(\vec{x}) = \sum_{i=1}^{n} i x_i^2$	$x_i \in [-10,10]$	0
$f_4(\vec{x}) = \sum_{i=1}^{n} \| x_i^2 \| + \prod_{i=1}^{n} \| x_i \|$	$x_i \in [-10,10]$	0
$f_5(\vec{x}) = \max\{\| x_i \|, 1 \le i \le n\}$	$x_i \in [-100,100]$	0
$f_6(\vec{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	$x_i \in [-100,100]$	0
$f_7(\vec{x}) = \sum_{i=1}^{n} i x_i^4$	$x_i \in [-1.28,1.28]$	0
$f_8(\vec{x}) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	$x_i \in [-1.28,1.28]$	0
$f_9(\vec{x}) = \sum_{i=1}^{n} z_i^2, \quad \vec{z} = \vec{x} - \vec{o}$	$x_i \in [-100,100]$	0
$f_{10}(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$x_i \in [-5.12,5.12]$	0
$f_{11}(\vec{x}) = \frac{1}{400} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$x_i \in [-600,600]$	0
$f_{12}(\vec{x}) = \sum_{i=1}^{n} \| x_i \|^{(i+1)}$	$x_i \in [-10,10]$	0
$f_{13}(\vec{x}) = -20\exp(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	$x_i \in [-32,32]$	0
$f_{14}(\vec{x}) = \frac{\pi}{n}\left\{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_i - 1)^2\right\} +$ $\sum_{i=1}^{n} u(x_i, 10, 100, 4) \;\; y_i = 1 + \frac{1}{4}(x_i + 1), \;\; u_{x_i,\alpha,k,m} = \begin{cases} k(x_i - \alpha)^m & x_i > \alpha \\ 0 & -\alpha \le x_i \le \alpha \\ k(x_i - \alpha)^m & x_i > \alpha \end{cases}$	$x_i \in [-50,50]$	0
$f_{15}(\vec{x}) = \frac{1}{10}\left\{\sin^2(\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{i+1})]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	$x_i \in [-50,50]$	0
$f_{16}(\vec{x}) = \sum_{i=1}^{n} \| x_i \cdot \sin(x_i) + 0.1 \times x_i \|$	$x_i \in [-10,10]$	0
$f_{17}(\vec{x}) = \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_{i+1}) + \| x_n - 1 \| [1 + \sin^2(3\pi x_n)]$	$x_i \in [-10,10]$	0
$f_{18}(\vec{x}) = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^{n} x_i^2}) - 0.5}{(1 + 0.001 \times \| \sum_{i=1}^{n} x_i^2 \|)^2}$	$x_i \in [-100,100]$	0
$f_{19}(\vec{x}) = \sum_{i=1}^{n} [x_i^2 - 100\cos(2\pi x_i) + 10]$	$x_i \in [-5.12,5.12]$	0

Table 2. Test results of high dimensional single-mode function.

Functions	Result	GPIO	PIO	IPIO	FOA	PSO
f_1	best	4.83E-57	1.87E-07	4.17E-15	4.21E-09	2.24E+00
	worst	1.60E-47	1.90E-05	1.34E-01	4.24E-09	2.90E+01
	mean	5.47E-49	4.03E-06	6.98E-03	4.22E-09	9.88E+00
	std	2.92E-48	5.20E-06	2.67E-02	7.85E-12	5.56E+00
	timec	4.61E-01	2.94E-01	5.98E-01	1.08E+00	8.13E-01
f_2	best	5.59E-53	1.21E-03	4.02E-06	2.49E-04	1.53E+04
	worst	7.82E-46	3.20E-01	6.99E+04	2.53E-04	2.24E+05
	mean	5.41E-47	3.19E-02	3.40E+03	2.50E-04	8.09E+04
	std	1.56E-46	6.93E-02	1.37E+04	7.98E-07	4.98E+04
	timec	1.39E+00	9.52E-01	1.97E+00	2.97E+00	1.31E+00
f_3	best	4.54E-56	1.33E-07	1.63E-10	7.30E-06	1.47E+00
	worst	4.16E-50	9.86E-06	7.63E-02	7.40E-06	1.65E+01
	mean	2.20E-51	1.43E-06	2.65E-03	7.35E-06	6.34E+00
	std	7.80E-51	1.86E-06	1.39E-02	2.17E-08	3.55E+00
	timec	4.36E-01	2.70E-01	5.51E-01	9.95E-01	7.45E-01
f_4	best	3.58E-28	5.90E-04	9.07E-07	3.84E-03	1.35E+00
	worst	1.88E-25	5.05E-03	2.04E+00	3.86E-03	7.66E+00
	mean	2.39E-26	2.15E-03	7.86E-02	3.85E-03	4.61E+00
	std	4.18E-26	1.19E-03	3.72E-01	4.52E-06	1.53E+00
	timec	5.02E-01	3.22E-01	6.53E-01	1.13E+00	8.16E-01
f_5	best	1.17E-27	5.70E-04	8.53E-07	1.30E-05	1.97E+00
	worst	8.76E-24	5.65E-03	1.29E+00	1.32E-05	4.85E+00
	mean	5.60E-25	2.00E-03	5.50E-02	1.31E-05	3.36E+00
	std	1.63E-24	1.17E-03	2.35E-01	3.80E-08	7.41E-01
	timec	4.44E-01	3.06E-01	6.20E-01	1.07E+00	7.58E-01
f_6	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.00E+00
	worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.00E+01
	mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.09E+01
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.97E+01
	timec	4.49E-01	2.91E-01	6.09E-01	1.08E+00	8.17E-01
f_7	best	2.16E-117	3.11E-17	2.02E-30	2.29E-08	1.23E-05
	worst	3.21E-99	1.20E-13	5.69E-05	4.62E-08	4.32E-03
	mean	2.01E-100	1.21E-14	2.82E-06	3.03E-08	4.61E-04
	std	7.60E-100	2.83E-14	1.14E-05	5.87E-09	8.11E-04
	timec	1.45E+00	9.84E-01	1.97E+00	2.92E+00	1.28E+00
f_8	best	8.19E-06	2.00E-03	8.03E-03	8.26E-04	4.99E-02
	worst	3.49E-03	6.54E-01	9.44E-01	3.73E-03	3.60E-01
	mean	9.68E-04	1.74E-01	1.67E-01	1.64E-03	1.49E-01
	std	1.04E-03	1.92E-01	2.66E-01	7.47E-04	6.06E-02
	timec	1.40E+00	9.98E-01	2.01E+00	2.98E+00	1.32E+00
f_9	best	3.35E-56	4.89E-07	3.81E-11	4.20E-09	1.39E+00
	worst	1.83E-48	2.93E-05	3.65E+01	4.23E-09	2.54E+01
	mean	8.44E-50	6.05E-06	1.23E+00	4.22E-09	7.88E+00
	std	3.34E-49	6.67E-06	6.66E+00	8.49E-12	6.25E+00
	timec	6.89E-01	4.79E-01	9.78E-01	1.60E+00	9.19E-01

According to the data shown in Table 2, in the solution of high-dimensional single-mode function, the optimal value, worst value, variance and average value obtained by GPIO algorithm after 30 iterations are better than those obtained by other algorithms. Therefore, it can be concluded that the convergence ability and solution accuracy of GPIO algorithm are better than those of other algorithms.

Table 3. Test results of High-dimensional Multimode function.

Functions	Resunt	OPIO	PIO	IPIO	FOA	PSO
f_{10}	best	0.00E+00	3.06E-05	5.32E-09	4.47E-04	2.00E+01
	worst	0.00E+00	6.04E-03	3.87E+00	4.51E-04	9.63E+01
	mean	0.00E+00	4.32E-04	2.55E-01	4.48E-04	5.47E+01
	std	0.00E+00	1.07E-03	9.53E-01	1.06E-06	1.82E+01
	timec	5.79E-01	4.14E-01	8.42E-01	1.33E+00	8.86E-01
f_{11}	best	0.00E+00	1.09E-06	2.80E-10	6.76E-12	1.02E+01
	worst	0.00E+00	5.27E-05	3.64E+00	6.82E-12	1.75E+01
	mean	0.00E+00	1.17E-05	2.23E-01	6.78E-12	1.35E+01
	std	0.00E+00	1.12E-05	7.06E-01	1.48E-14	2.23E+00
	timec	7.97E-01	5.48E-01	1.12E+00	1.73E+00	1.13E+00
f_{12}	best	4.39E-60	8.24E-11	5.86E-15	6.09E-09	2.68E-05
	worst	2.20E-52	3.60E-07	4.78E-04	2.66E-05	1.13E+01
	mean	8.52E-54	6.72E-08	3.40E-05	8.92E-07	9.81E-01
	std	4.02E-53	9.15E-08	1.10E-04	4.85E-06	2.66E+00
	timec	1.37E+00	9.30E-01	1.88E+00	2.84E+00	1.26E+00
f_{13}	best	8.88E-16	3.15E-04	3.38E-07	1.50E-04	2.07E+00
	worst	8.88E-16	3.81E-03	5.30E-01	1.51E-04	5.05E+00
	mean	8.88E-16	1.06E-03	3.57E-02	1.51E-04	3.31E+00
	std	0.00E+00	8.41E-04	1.05E-01	2.08E-07	6.58E-01
	timec	5.86E-01	4.25E-01	8.78E-01	1.41E+00	8.60E-01
f_{14}	best	1.09E-01	8.69E-01	2.32E-01	1.15E+00	1.03E+00
	worst	1.15E+00	1.15E+00	6.98E+03	1.15E+00	7.24E+00
	mean	4.82E-01	1.13E+00	2.34E+02	1.15E+00	3.98E+00
	std	3.09E-01	5.11E-02	1.27E+03	1.05E-07	1.54E+00
	timec	6.65E-01	3.91E-01	8.44E-01	1.38E+00	8.83E-01
f_{15}	best	8.41E-01	2.18E+00	2.47E+00	2.75E+00	2.55E+00
	worst	2.90E+00	2.90E+00	3.17E+00	2.83E+00	2.80E+01
	mean	2.10E+00	2.87E+00	2.89E+00	2.80E+00	1.42E+01
	std	6.67E-01	1.34E-01	1.34E-01	2.10E-02	6.91E+00
	timec	6.99E-01	6.13E-01	1.43E+00	1.94E+00	1.01E+00
f_{16}	best	9.76E-29	1.12E-04	2.32E-06	3.84E-04	9.82E-02
	worst	2.58E-24	1.87E-03	1.33E+01	3.86E-04	5.17E+00
	mean	2.97E-25	6.15E-04	4.48E-01	3.85E-04	1.92E+00
	std	6.32E-25	3.72E-04	2.42E+00	4.29E-07	1.38E+00
	timec	4.94E-01	3.30E-01	6.73E-01	1.15E+00	7.99E-01
f_{17}	best	7.07E-06	9.41E+00	1.70E+01	2.71E+01	7.64E-01
	worst	1.12E+01	2.90E+01	3.02E+01	2.87E+01	8.74E+00
	mean	5.37E+00	1.91E+01	2.52E+01	2.79E+01	3.61E+00
	std	2.91E+00	6.10E+00	4.23E+00	3.03E-01	2.07E+00
	timec	5.20E-01	3.79E-01	7.86E-01	1.28E+00	8.29E-01
f_{18}	best	0.00E+00	1.61E-03	1.12E-11	4.20E-09	1.78E-01
	worst	0.00E+00	3.75E-02	3.51E-02	4.23E-09	3.12E-01
	mean	0.00E+00	1.12E-02	4.60E-03	4.22E-09	2.51E-01
	std	0.00E+00	5.95E-03	8.39E-03	8.01E-12	3.68E-02
	timec	4.99E-01	3.20E-01	6.50E-01	1.17E+00	8.99E-01
f_{19}	best	0.00E+00	1.31E-05	1.20E-11	4.46E-04	2.16E+01
	worst	0.00E+00	2.18E-03	1.52E+02	4.51E-04	7.73E+01
	mean	0.00E+00	2.81E-04	1.02E+01	4.48E-04	4.35E+01
	std	0.00E+00	3.99E-04	3.77E+01	1.11E-06	1.42E+01
	timec	5.25E-01	3.77E-01	7.71E-01	1.27E+00	8.71E-01

The optimization time of GPIO algorithm is longer than that of traditional pigeon swarm algorithm, but the optimal value obtained by GPIO algorithm is several times or dozens of times higher than that of PIO algorithm. Compared with IPIO algorithm, FOA algorithm and PSO algorithm improved in literature [14], the data obtained by the improved algorithm in this paper is better in terms of time and other data, the results show that GPIO algorithm has good optimization performance and strong stability in solving high-dimensional single-mode functions. In solving the high-dimensional multimode function problem, this paper uses 10 test functions, and the results are shown in Table 3.

It can be seen from table 3 that the optimal value, worst value, average value and variance obtained by GPIO algorithm for solving high-dimensional multimode function are better than the other four algorithms.

In order to more intuitively understand the change of the optimal value of GPIO algorithm, PIO algorithm, ipio algorithm, FOA algorithm and PSO algorithm in the optimization process, this paper gives the change curves of function fitness value solved by five algorithms, as shown in Figure 1. Due to the limitation of the length of the paper, we only give the convergence graphs of six functions.

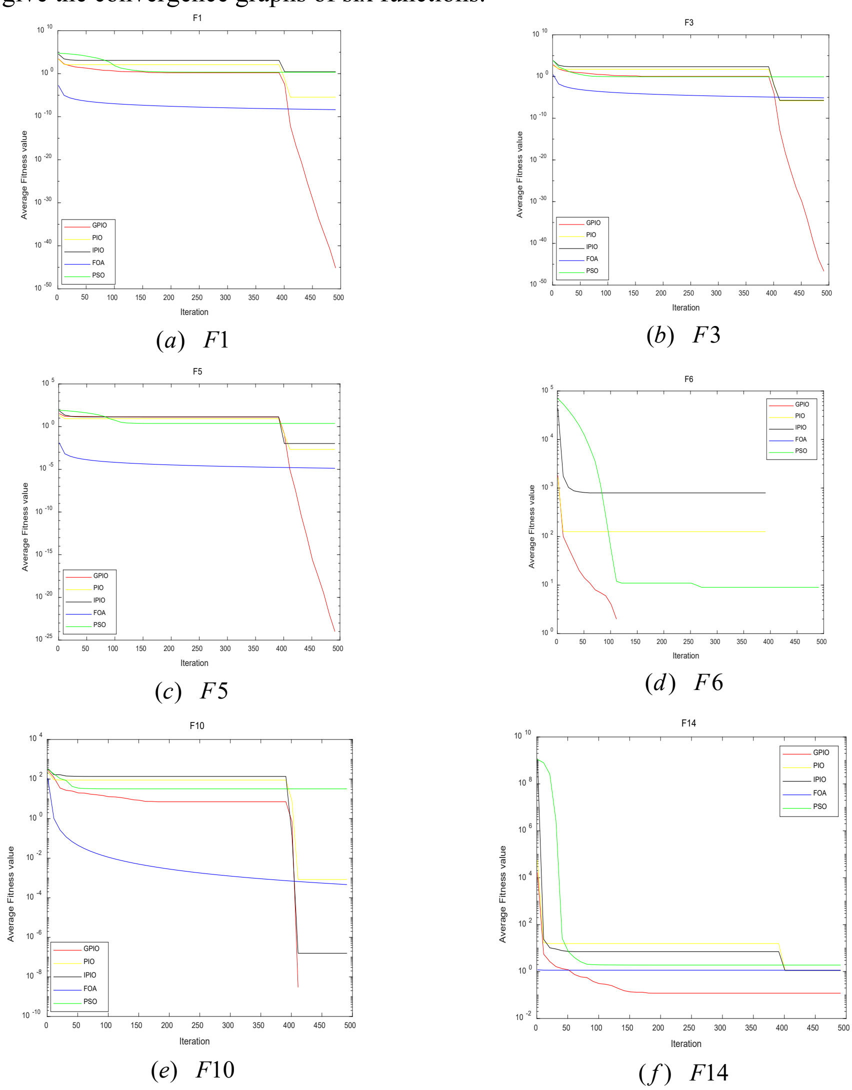

Fig. 1. Suitability value change curve of partial functions.

According to figure 1, the convergence accuracy and global search ability of GPIO algorithm are better than the other four algorithms. Among them, PIO algorithm, ipio algorithm, FOA algorithm and POS algorithm fall into local optimization in the later stage of iteration, while GPIO algorithm has searched the theoretical optimal solution 0 before the end of iteration. Since it is meaningless to take the logarithm of 0, Therefore, the fitness curve is interrupted when the algorithm searches for the optimal solution 0 without reaching the maximum number of iterations.

5 Conclusions

Aiming at the problems that the pigeon swarm optimization algorithm is easy to fall into local optimization and the convergence accuracy is not high in the later stage of iteration, this paper calls the cubic mapping method to increase the diversity of the population; The Gaussian mutation idea makes it easier for the pigeon swarm algorithm to call out the local optimal solution, enhances the global search ability of the algorithm, and effectively balances the local and local search ability of the algorithm; The decreasing factor is used to improve the convergence accuracy of the algorithm in the later stage of iteration. Through the simulation experiments of 19 test functions and power system economic dispatching, the experimental results show that the GPIO algorithm proposed in this paper has better global search ability, robustness, convergence speed and convergence accuracy than other swarm intelligence optimization algorithms.

This work was supported by the Basic Ability Improvement Project for Young and Middle-Aged Teachers in Colleges and Universities in Guangxi (no. 2021KY1251).

References

1. Sayed, Gehad Ismail, et al. “Feature Selection via a Novel Chaotic Crow Search Algorithm.” Neural Computing and Applications, vol. 31, no. 1, 2019, pp. 171–188.
2. Kennedy, J., and R. Eberhart. “Particle Swarm Optimization.” Neural Networks, 1995. Proceedings., IEEE International Conference On, vol. 4, 2002, pp. 1942–1948.
3. Yang, Xin-She, and Xingshi He. “Bat Algorithm: Literature Review and Applications.” International Journal of Bio-Inspired Computation, vol. 5, no. 3, 2013, pp. 141–149.
4. Arora, Sankalap, and Satvir Singh. “Butterfly Optimization Algorithm: A Novel Approach for Global Optimization.” Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive, vol. 23, no. 3, 2019, pp. 715–734.
5. Yang, Xin-She. “Flower Pollination Algorithm for Global Optimization.” UCNC’12 Proceedings of the 11th International Conference on Unconventional Computation and Natural Computation, 2012, pp. 240–249.
6. Duan, Haibin, and Peixin Qiao. “Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning.” International Journal of Intelligent Computing and Cybernetics, vol. 7, no. 1, 2014, pp. 24–37.
7. Mirjalili, Seyedali, and Andrew Lewis. “The Whale Optimization Algorithm.” Advances in Engineering Software, vol. 95, no. 95, 2016, pp. 51–67.
8. Mirjalili, Seyedali, et al. “Grey Wolf Optimizer.” Advances in Engineering Software, vol. 69, 2014, pp. 46–61.
9. Rao, R.Venkata. Teaching–Learning-Based Optimization Algorithm. 2016, pp. 211–216.

10. Duan Haibin, Ye Fei. Progress in Pigeon Cluster Optimization Algorithm. Journal of Beijing University of Technology,2017, 43(01):1-7.
11. [11] Tao Guojiao,Li Zhi. Journal of SichuanUniversity (Natural Edition) 2018, 55 Science (02):295-300. (in Chinese).
12. Ji Xiaobo, Jin Jingfeng, Li Xianfeng. Improved Pigeon Group Algorithm for Solving High -dimensional ComplexFunctions. Information & Computer (TheoreticalEdition), 2020, 32(24):44-47.
13. Shuanglin Li, Jiahao He, Haiyue Ao, Yanbin Liu. Real-time Obstacle-avoidance Algorithm Based on Pigeon-swarm Optimization Algorithm. Journal of Beijing University of Aeronautics and Astronautics: 1-9[2021-06-19].
14. Chen Zhonghua, Liu Bo, Guo Rui, Tang Jun. MPPT method for photovoltaic array based on improved pigeon swarm algorithm. Journal of Power System and Automation, 1-10[2021-06-15].
15. Guilford Tim,Roberts Stephen,Biro Dora,Rezek Iead. Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models. Journal of theoretical biology, 2004, 227(1).
16. Victoria A. Braithwaite,Tim Guilford. Viewing Familiar Landscapes Affects Pigeon Homing. Proceedings of the Royal Society B: Biological Sciences, 1991, 245(1314).
17. Xiaoyu Song,Ming Zhao,Qifeng Yan,Shuangyun Xing. A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization. Swarm and Evolutionary Computation, 2019, 50.
18. Wang Lin, Lv Shengxiang, Zeng Yurong. A Review of Optimization Algorithms for Drosophila. Control and Decision, 2017, 32(07):1153-1162.
19. Yang Wei, Li Qiqiang. Review of Particle Swarm Optimization Algorithms. Engineering Science, 2004(05):87-94.