

Received May 29, 2021, accepted June 15, 2021, date of publication June 18, 2021, date of current version June 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3090512

# Flamingo Search Algorithm: A New Swarm Intelligence Optimization Algorithm

WANG ZHIHENG<sup>ID</sup> AND LIU JIANHUA<sup>ID</sup>

School of Computing, Xi'an University of Posts and Telecommunications, Xi'an 710100, China

Corresponding author: Wang Zhiheng (wzh3147255630@163.com)

**ABSTRACT** This paper presents a new swarm intelligence optimization algorithm: Flamingo Search Algorithm (FSA), which is inspired by the migratory and foraging behavior of flamingos. A mathematical model of flamingo behavior is built so that FSA has the global exploration and local exploitation capabilities required for an optimization algorithm. Three sets of experiments based on 68 test functions are designed to evaluate the convergence speed, optimization-seeking accuracy, stability, running time, and global search capability of FSA. The effect of different input parameters on the search results of FSA is then discussed, and the optimal parameter selection interval is summarized. In addition, nine test functions are selected to visualize the trajectory of the flamingo population during the search. The test results of the above designs all indicate that FSA is superior to other algorithms in solving optimization problems. Finally, three kinds of simulation experiments, which are push-pull circuit problem, path planning problem and network intrusion detection system, are designed to test the practicability of FSA. The code used for the main experiment in this article can be obtained from website <https://github.com/18280426650/FSA>.

**INDEX TERMS** Flamingo search algorithm (FSA), bio-inspired algorithms, optimization problems, intrusion detection system, intelligence optimization algorithms.

## I. INTRODUCTION

The optimization of specific parameters in engineering problems has been a hot topic of research in related fields. According to the way of finding and characteristics, optimization algorithms can be divided into two main categories: deterministic algorithms and stochastic algorithms [1]. The common deterministic algorithms include the steepest descent method [2], newton method [3], and conjugate gradient methods [4]. These algorithms are more mature in mathematical theory, but they have serious defects. When the objective function is discontinuous, non-derivative and non-convex, the convergence speed and local convergence problems easily occur when using these algorithms, leading to poor optimization and performance. However, most of the problems in practical engineering applications are accompanied by a large number of local solutions, and these problems are often difficult to quantify as a continuous, derivable function that exhibits convexity, resulting in the limitations of deterministic algorithms. Consequently, academic circles researched stochastic optimization algorithm

comprehensively. Compared with deterministic algorithm, stochastic optimization algorithm has the main feature of greater randomness. A kind of the typical stochastic optimization, swarm intelligence algorithm finds a path with good convergence speed and convergence performance to solve an optimization problem in a unique way [5]. Swarm intelligence optimization algorithms are easy to implement and adaptable and have certain stability, robustness, and scalability [6]. Moreover, the population as a whole requires relatively little supervision or top-down control.

In recent years, many new swarm intelligence optimization algorithms have been proposed, such as Ant Colony Optimization (ACO) [7], [8], Particle Swarm Optimization (PSO) [9], [10], Cuckoo Search (CS) [11], [12], Grey Wolf Optimization (GWO) [13], [14], Whale Optimization Algorithm (WOA) [15], Tunicate Swarm Algorithm (TSA) [16], Sparrow Search Algorithm (SSA), and so on [17]. All these algorithms have the advantages of high applicability, low parameters, and high global search capability. They are widely used in solving practical engineering design optimization problems.

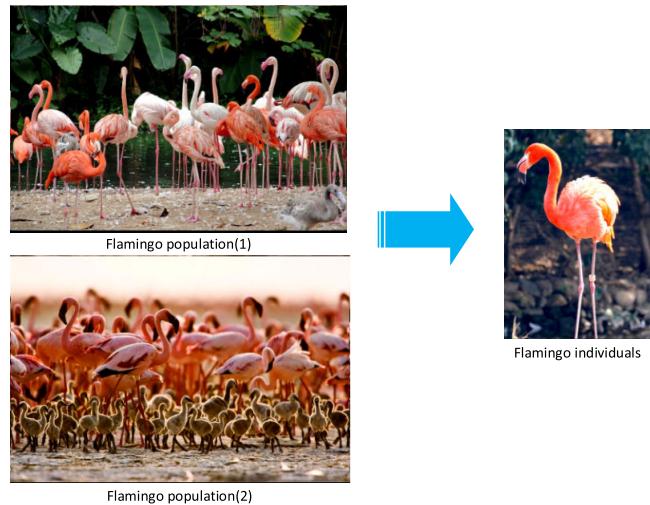
PSO is inspired by the behavior of bird populations, and the core idea is to simulate the behavioral characteristics of a

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang<sup>ID</sup>.

flock of birds when searching for food, with each bird representing a particle in the PSO and the corresponding particle adaptation value determined by the objective function[18]. ACO is inspired by the foraging behavior of ant populations. In the process of searching for food, ants leave behind a substance called pheromone, which becomes less and less as the distance the ants travel increases, and each ant decides which way to go based on the concentration of pheromones around it. The optimal solution of the problem is found through the constant updating of the ant position [19]. SSA simulates the behavioral characteristics of a sparrow population and abstracts the behavior of the sparrow population into a discoverer model, a joiner model, and an early-warning model. The discoverer consists of the sparrow with the best fitness, which leads the search of the whole population; the joiner follows the discoverer in a random search; and the early warning is to solve the local convergence problem [20]. The interplay of the above three modules guide the entire population towards the optimal solution of the problem.

With the development of engineering technology, complex engineering problems become more and more frequent, which leads to the desire to expand the diversity of swarm intelligence algorithm and improve its performance. But the core of improving the performance of algorithm optimization is to increase the ability to explore and develop the search space, and to ensure a good balance between exploration and development. The exploration ability of the algorithm refers to the examination of the promising area in the search space, while the exploitation refers to the search for the optimal solution in the promising area [21]. We need to find a breakthrough between the two, but at the same time we need to ensure the balance between the two. But new swarm intelligence algorithms have been proposed in recent years, so why do we need to develop more optimization techniques? This problem is well explained by the no free lunch (NFL) theorem [22], which states that, for any algorithm, the improvement in performance on one kind of problem is offset by the improvement in performance on another. That is, the performance of an optimization algorithm may degrade after solving other problems of a different nature. NFL theorem states that researchers need to come up with new intelligent algorithms to solve optimization problems in different application areas. This also prompted us to develop a new effective optimization algorithm to solve nonlinear optimization problems.

This paper introduces a new swarm intelligence optimization technique in detail, that is, Flamingo Search Algorithm (FSA). FSA is a novel swarm intelligence optimization algorithm inspired by the migratory and foraging behavior of flamingos. In Section II, the construction of flamingo population model is described along with the specific ideas of FSA. In Sections III, IV, and V, this algorithm is evaluated in terms of search accuracy, robustness, convergence speed, and operational efficiency using benchmarking functions. Finally, in Section VI, the algorithm is tested in three specific engineering applications in three different domains.



**FIGURE 1. Flamingo population.**

## II. FLAMINGO SEARCH ALGORITHM

### A. FLAMINGO CHARACTERISTICS

Flamingos are gregarious migratory birds that feed mainly on algae, small shrimps, clams, small worms and insect larvae. Flamingos also feed in a distinctive manner, by bending their long necks down and turning their heads over, then walking while sweeping their curved beaks around their bodies and touching the bottom of the water to feed [23]. Flamingo populations and individuals in nature are shown in Fig. 1.

The two main behavioral characteristics of flamingos are foraging and migratory behavior. Flamingo populations mainly inhabit areas where food is plentiful. After a period of extensive foraging, flamingo populations migrate when the food in the area is reduced to a level that cannot satisfy the population [24]. This paper establishes the corresponding foraging model and migration model. The main optimization ideas of FSA model are shown as follows.

1. Flamingos sing to each other to communicate their location, as well as the availability of food in their location.

2. The population of flamingos does not know where the most food is in the current search area. Instead, they only update the location of each flamingo (which is affected by foraging behavior and migration behavior) to find the location of more food than the known food in the search area. This behavior of flamingos just fits the optimization idea of swarm intelligence algorithm: to seek the global optimal solution in a certain search space. While the search agent is doing a search, we are unable to know the current global optimal solution in search space is how much, search agent is the flamingos in the FSA, flamingos explore the search space and development through information exchange between each other and fixed rules of location movement, which ultimately leads to the optimal solution.

3. The rules of position change are based on the behavior of the flamingos. Its main behavior mainly has two kinds: foraging behavior and migration behavior. Foraging behavior can be divided into two behavioral characteristics:

beak scanning and foot movement, both of which affect the foraging behavior of flamingos.

### B. BASIC IDEAS OF THE ALGORITHM

The FSA's mathematical model is described below.

#### 1) FORAGING BEHAVIOR

Feature 1: Communicative behavior.

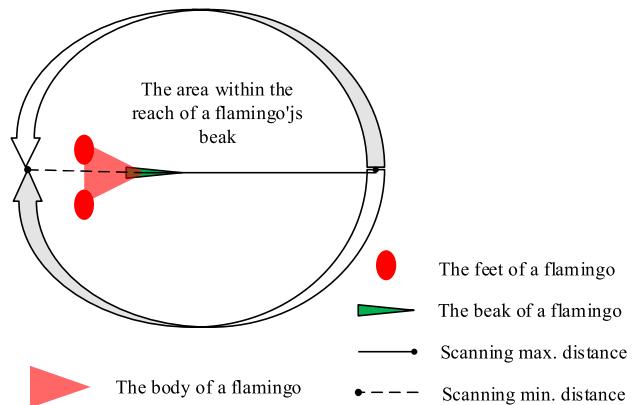
The flamingos that have the most food in the group call other flamingos to spread their location information and influence the position changes of other flamingos in the group. For a population of flamingos, the location where the most food is known is where the flamingos that have the most food in the population are located. In theory, flamingos have no way of knowing where the most food is in an area (the global optimal). However, this does not mean that the algorithm cannot find the global optimal, because the algorithm is a kind of program, and we cannot tell the ending condition of the program at the time of setting it.

FSA is an algorithm that simulates flamingos trying to find the optimal solution in the search area (i.e. the location of the food is most abundant) based on the limited information available. This paper assumes that the flamingo that has the most food in the  $j$ th dimension is  $xb_j$ .

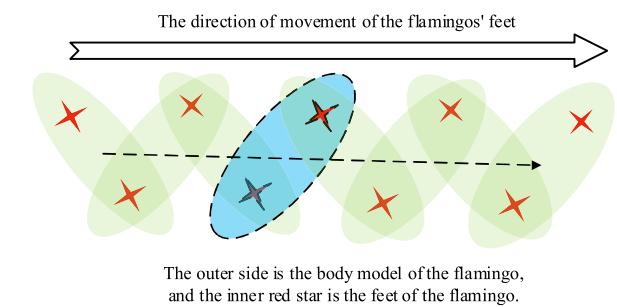
Feature 2: Beak scanning behavior.

A flamingo's bill, when inverted in water, acts like a large sieve, sucking water in and filtering it out quickly because of the deep grooves in the lower bill of the flamingo and the shallow, capped grooves in the upper bill, with sparse serrations and fine hairs around the edges. When flamingos forage for food, their heads dip downward, their mouths turn upside down, and they eat food into their mouths, discharging excess water and inedible dregs. This way of foraging is influenced by the abundance of food in the area. If the area swept by a flamingo's beak is richer in food, this encourages the flamingo to scan the area more carefully, and the flamingo's neck will slowly stretch out, causing the beak to increase its scanning radius. The probability of scanning the area for food also increases. The beak scanning behavior model of a flamingo is shown in Fig. 2.

The closer a flamingo is to the location of the most abundant food in the population, the more likely there will be abundant food in that area. This paper simulates the beak scanning behavior of flamingos. Supposing the position of the  $i$ th flamingo in the  $j$ th dimension of the flamingo population is  $x_{ij}$  and taking into account the variability of each individual flamingo's choice in nature and the influence of the suddenness of the specific environment on the flamingo's foraging behavior, the foraging behavior of individual flamingos encounters an error with the information transmitted. To simulate this error, a standard normal random distribution is introduced, in which the beak scan of a flamingo has a high probability of being aligned with the direction of the location where food is most abundant. However, there is also a small probability of errors based on this information.



**FIGURE 2. Beak scanning behavior.**



**FIGURE 3. Bipedal mobile behavior.**

Then the maximum distance of the flamingo's beak scan in foraging behavior can be quantified as  $|G_1 \times xb_j + \varepsilon_2 \times x_{ij}|$ , where  $\varepsilon_2$  is a random number of  $-1$  or  $1$ . The maximum distance is primarily intended to increase the search range of the flamingo's beak scan in its foraging mode, where  $G_1$  is a random number that follows a standard normal distribution. To simulate the scanning range of flamingos during beak scanning behavior, the normal distribution is again introduced, and its variation curve approximates the variation of the flamingo's beak scanning range as  $G_2 \times |G_1 \times xb_j + \varepsilon_2 \times x_{ij}|$ , where  $G_2$  is a random number that obeys the standard normal distribution.

Feature 3: Bipedal mobile behavior.

The foot movement behavior model of flamingos is shown in Fig. 3. When flamingos forage, while scanning with their beaks, their claws move toward where food is most abundant in the flamingo population. Assuming the location where food is most abundant in the population is  $xb_j$ , the distance traveled can be quantified as  $\varepsilon_1 \times xb_j$ , where  $\varepsilon_1$  is a random number of  $-1$  or  $1$ , which is mainly to increase the search range of flamingos foraging and quantify individual differences in choice.

To sum up, the moving step of flamingos foraging in the  $t$ th iteration is the scanning range of flamingo beak plus the moving distance of their feet, as shown in (1).

$$b_{ij}^t = \varepsilon_1 \times xb_j^t + G_2 \times |G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t| \quad (1)$$

The equation for updating the location of flamingo foraging behavior is

$$x_{ij}^{t+1} = (x_{ij}^t + \varepsilon_1 \times xb_j^t + G_2 \times |G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t|)/K \quad (2)$$

In (2),  $x_{ij}^{t+1}$  represents the position of the  $i$ th flamingo in the  $j$ th dimension of the population in the  $(t+1)$ th iteration,  $x_{ij}^t$  represents the position of the  $i$ th flamingo in the  $j$ th dimension in the  $t$  iteration of the flamingo population, namely, the position of the flamingo's feet.  $xb_j^t$  represents the  $j$ th dimension position of the flamingo with the best fitness in the population in the  $t$  iteration.  $K = K(n)$  is a diffusion factor, which is a random number following the chi-square distribution of  $n$  degrees of freedom. It is used to increase the size of the flamingo's foraging range and to simulate the chance of individual selection in nature, increasing its global merit-seeking ability.  $G_1 = N(0, 1)$  and  $G_2 = N(0, 1)$  are random numbers that follow a standard normal distribution,  $\varepsilon_1$  and  $\varepsilon_2$  are randomized by  $-1$  or  $1$ .

## 2) MIGRATION BEHAVIOR

When food is scarce in the present foraging area, the flamingo population migrates towards the next area where food is more abundant. Assuming that the location of the food-rich area in the  $j$ th dimension is  $xb_j$ , the formula for the migration of the flamingo population is as follows.

$$x_{ij}^{t+1} = x_{ij}^t + \omega \times (xb_j^t - x_{ij}^t) \quad (3)$$

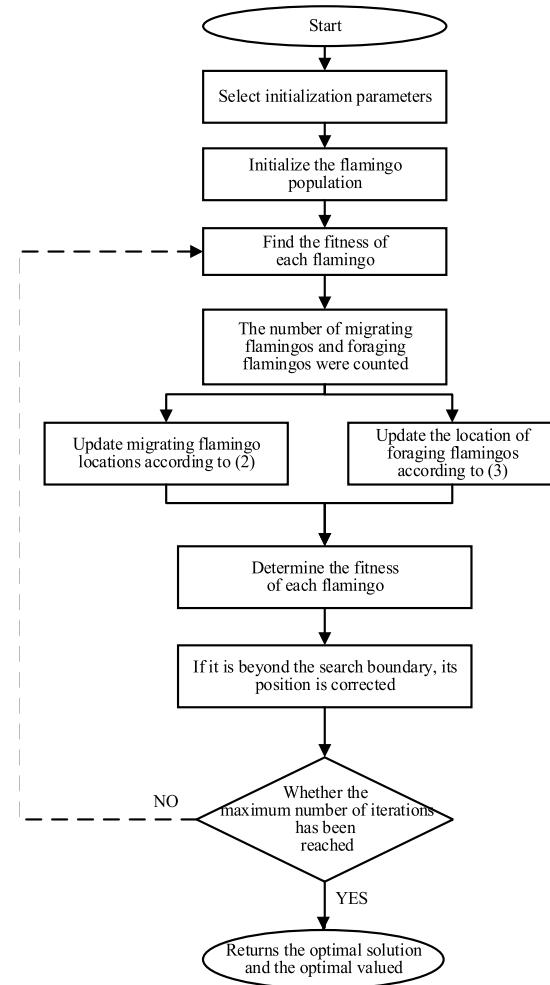
In (3),  $x_{ij}^{t+1}$  represents the position of the  $i$ th flamingo in the  $j$ th dimension of the population in the  $t+1$  iteration, and  $x_{ij}^t$  represents the position of the  $i$ th flamingo in the  $j$ th dimension in the  $t$  iteration of the flamingo population, namely, the position of the flamingo's feet.  $xb_j^t$  represents the  $j$ th dimension position of the flamingo with the best fitness in the population in the  $t$  iteration.  $\omega = N(0, n)$  is a Gaussian random number with  $n$  degrees of freedom, which is used to increase the search space during the migration of flamingos and simulate the randomness of individual behaviors of flamingos in the specific migration process.

## C. ALGORITHM FLOW

This sub-section sets out the basic process of FSA.

*Step1:* The population is initialized, the population is set as  $P$ , the maximum number of iterations is  $Iter_{Max}$ , and the proportion of migrating flamingos in the first part is  $MP_b$ .

*Step2:* The number of foraging flamingos in the  $i$ th iteration of flamingo population renewal is  $MP_r = rand [0,1] \times P \times (1-MP_b)$ . The number of migrating flamingos in the first part of this iteration is  $MP_o = MP_b \times P$ . The number of migratory flamingos in the second part of this iteration is  $MP_t = P - MP_o - MP_r$ . The fitness values of individual flamingos are obtained, and the population of flamingos is sorted according to the fitness values of individual flamingos. The former flamingos  $MP_b$  with low fitness and the former flamingos  $MP_t$  with high fitness are regarded as



**FIGURE 4. Flow chart of FSA.**

migratory flamingos, whereas the others are regarded as foraging flamingos.

*Step3:* Migrating flamingos are updated according to (3), and foraging flamingos are updated according to (2).

*Step4:* Check for flamingos that are out of bounds (see pseudocode for details).

*Step5:* If the maximum number of iterations is reached, go to Step6; otherwise, go to Step2.

*Step6:* Output the optimal solution and optimal value.

The flow chart of FSA is shown in Fig. 4.

## D. ALGORITHM PSEUDO-CODE

This sub-section shows pseudo-code of FSA.

## E. COMPLEXITY ANALYSIS OF FSA

This subsection analyzes the complexity of the FSA. The time complexity and space complexity of FSA are described separately below.

### 1) TIME COMPLEXITY

Initializing the population takes  $O(n \times d)$  time, where  $n$  is the population size and  $d$  is the dimension size. The

**Algorithm 1** Flamingo Search Algorithm

---

```

Input: Flamingo population size,  $P$ ;
Maximum number of iterations,  $IterMax$ ;
The first part is the proportion of migrating flamingos,
 $MP_b$ ;
Output: The optimal fitness value,  $f_g$ ; The optimal solution,
 $x_{best}$ ;
1: Rank the fitness values and find the current best individual,
 $x_{best}$ ;
2:  $t \leftarrow 1$ .
3: while ( $t \leq IterMax$ )do
4:    $R \leftarrow rand[0,1]$ .
5:    $MP_r \leftarrow R \times P \times (1 - MP_b)$ .
6:    $MP_0 \leftarrow MP_b$ .
7:    $MP_t \leftarrow P - MP_0 - MP_r$ .
8:   for  $i \leftarrow 1$  to  $MP_b$ do
9:     for  $j \leftarrow 1$  to  $n$  do //n is the dimension size
10:      Using (3) update the flamingo's location;
11:    end for
12:   end for
13:   for  $i \leftarrow 1 + MP_0$  to  $MP_0 + MP_r$  do
14:     for  $j \leftarrow 1$  to  $n$  do
15:       Using (2) update the flamingo's location;
16:     end for
17:   end for
18:   for  $i \leftarrow MP_0 + MP_r + 1$  to  $P$  do
19:     for  $j \leftarrow 1$  to  $n$  do
20:       Using (3) update the flamingo's location;
21:     end for
22:   end for
23:   for  $i \leftarrow 1$  to  $P$  do //Boundary detection;
24:     for  $j \leftarrow 1$  to  $d$  do
25:       if  $x_{ij}^t > ub$  then
26:          $x_{ij}^t \leftarrow ub$ 
27:       end if
28:       if  $x_{ij}^t < lb$  then
29:          $x_{ij}^t \leftarrow lb$ 
30:       end if
31:     end for
32:   end for
33:   Rank the fitness values and find the current best
individual,  $x_{best}$ ;
34:    $t \leftarrow t + 1$ 
35: end while
36: return  $f_g$  and  $x_{best}$ 

```

---

$O(IterMax \times n \times d)$  time required to find the fitness of each flamingo, where  $IterMax$  is the maximum number of iterations. Migration and foraging take  $O(n \times d)$  time together.

## 2) SPACE COMPLEXITY

The spatial complexity of the FSA is  $O(n \times d)$ , where  $n$  is the population size and  $d$  is the dimension size, that is,

**TABLE 1.** Parameter settings for the five algorithms.

| Algorithm | Parameter               | Values           |
|-----------|-------------------------|------------------|
| PSO       | Inertia coefficient $w$ | 0.9              |
|           | Parameter $c_1$         | 2                |
|           | Parameter $c_2$         | 2                |
| WOA       | Control parameter $a_1$ | [2, 0]           |
|           | Control parameter $a_2$ | [-1, -2]         |
| GWO       | Control parameter $a$   | [2, 0]           |
|           | Parameter $P_{min}$     | 1                |
| TSA       | Parameter $P_{max}$     | 4                |
|           | Parameter $MP_b$        | 0.1              |
| FSA       | Diffusion factor        | $G(0,1.2), K(8)$ |

the maximum amount of space occupied when initializing the population.

**III. TEST SIMULATION OF THE BENCHMARK FUNCTION**

The experiments in this paper were all conducted on a computer with an Intel(R) Core (TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz, 16 GB of RAM, and Window 10 as the operating system. The code for the experiments was written and run in Python (Note: Section VI.B was written and run on MATLAB), with Python version 3.8.3, the development tool PyCharm version 2020.3.3, and the graphing tool MATLAB version 2018a.

In this subsection, three sets of test experiments are designed to test the convergence accuracy, robustness, and convergence speed of FSA for different types of test functions (single-peaked, multi-peaked, and fixed-dimensional). A brief analysis and summary of the test results are also presented. To form a comparison, four popular swarm intelligence optimization algorithms, namely, PSO, WOA, GWO, and TSA, are compared with FSA. PSO is an earlier proposed swarm intelligence algorithm, which has been widely applied in specific projects. The reason why WOA and GWO were chosen is that these two algorithms were proposed ten years ago. The same two algorithms have strong optimization performance and have been applied in engineering to a certain extent. TSA was chosen because this algorithm was just proposed last year, and its optimization performance could represent the latest research progress in the field of intelligent algorithms. When testing the specific optimization effect of FSA, the comprehensive comparison of these four population intelligent algorithms can evaluate the optimization effect of FSA algorithm more objectively, whether in theory or in concrete engineering practice.

The parameters of the five algorithms are set as follows.

To ensure the fairness of the comparison of the five algorithms, the maximum number of iterations is set to 300 and the number of populations is set to 50. The dimensional values of each test function are taken as shown in Tables 2, 4, and 6. As the swarm intelligence optimization algorithm is a stochastic algorithm, the convergence results of a single test inevitably have deviations. To avoid the effect of the randomness of the single test results on the test accuracy and facilitate the evaluation of its robustness, this

**TABLE 2.** Single-peak test functions.

| Function formula   | Region of search | Best | Dimensionality |
|--|------------------|------|----------------|
| $F_1(x) = \sum_{i=1}^n x_i^2$  | [-100,100]       | 0    | 20             |
| $F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $                        | [-10,10]         | 0    | 20             |
| $F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$                  | [-100,100]       | 0    | 20             |
| $F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$                               | [-100,100]       | 0    | 20             |
| $F_5(x) = \sum_{i=1}^{n-1} [100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | [-2.048,2.048]   | 0    | 20             |
| $F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$                                    | [-100,100]       | 0    | 20             |
| $F_7(x) = \sum_{i=1}^n i \times x_i^4 + \text{random}[0,1)$                | [-1.28,1.28]     | 0    | 20             |
| $F_8(x) = \sum_{i=1}^n  x_i ^{i+1}$  | [-1,1]           | 0    | 20             |
| $F_9(x) = \sum_{i=1}^n i x_i^2$  | [-10,10]         | 0    | 20             |

**TABLE 3.** Comparison of single-peak function tests.

| Algorithm | ATR     | F1               | F2               | F3               | F4              | F5              | F6              | F7              | F8               | F9               |
|-----------|---------|------------------|------------------|------------------|-----------------|-----------------|-----------------|-----------------|------------------|------------------|
| PSO       | Best    | 3.42E-01         | 2.20E+00         | 2.91E+02         | 6.09E-01        | 8.84E+01        | 3.23E-01        | 6.02E-02        | 1.78E-03         | 1.42E+00         |
|           | STD     | 2.75E-01         | 5.48E+00         | 6.59E+02         | 2.47E-01        | 3.02E+02        | 2.81E-01        | 1.84E-01        | 1.10E-02         | 3.53E+00         |
|           | AVG     | 2.23E+00         | 1.37E+01         | 3.56E+03         | 1.05E+00        | 9.53E+02        | 8.22E-01        | 3.12E-01        | 1.60E-02         | 6.27E+00         |
|           | Runtime | <b>1.0678 s</b>  | <b>1.5554 s</b>  | <b>1.5330 s</b>  | <b>1.1732 s</b> | <b>1.3880 s</b> | <b>1.1798 s</b> | <b>1.2193 s</b> | <b>1.3916 s</b>  | <b>1.1718 s</b>  |
| WOA       | Best    | 5.10E-21         | 7.76E-14         | 5.05E-01         | 3.04E-05        | 1.62E+01        | 3.19E-03        | 6.67E-05        | 5.48E-51         | 2.29E-22         |
|           | STD     | 2.71E-17         | 3.80E-12         | 4.54E+02         | 4.38E-04        | 7.98E-01        | 3.79E-01        | 3.52E-04        | 5.25E-38         | 2.84E-18         |
|           | AVG     | 4.21E-17         | 6.97E-12         | 8.40E+02         | 4.92E-04        | 5.18E+01        | 6.89E-01        | 5.00E-04        | 1.25E-38         | 8.68E-19         |
|           | Runtime | 1.3583 s         | 1.8901 s         | 1.7927 s         | 1.4872 s        | 1.6550 s        | 1.4885 s        | 1.5746 s        | 1.6941 s         | 1.4671 s         |
| GWO       | Best    | 2.08E-29         | 3.75E-17         | 2.62E+01         | 2.02E-08        | <b>1.60E+01</b> | <b>1.07E-05</b> | 2.69E-04        | 2.23E-65         | 3.61E-31         |
|           | STD     | 1.23E-27         | 1.12E-16         | 1.11E+03         | 1.46E-07        | 5.99E-01        | <b>1.89E-01</b> | 7.08E-04        | 2.73E-60         | 5.04E-29         |
|           | AVG     | 2.95E-27         | 5.19E-16         | 3.97E+03         | 1.39E-07        | <b>5.04E+01</b> | <b>1.50E-01</b> | 1.18E-03        | 8.60E-61         | 2.57E-29         |
|           | Runtime | 2.1069 s         | 3.1718 s         | 3.0203 s         | 2.1243 s        | 2.7184 s        | 2.3113 s        | 2.3481 s        | 2.7191 s         | 2.2806 s         |
| TSA       | Best    | 4.29E-71         | 2.33E-37         | 1.60E-48         | 6.69E-33        | 1.78E+01        | 2.41E+00        | 6.95E-06        | 6.48E-95         | 4.43E-73         |
|           | STD     | 2.28E-70         | 3.17E-37         | 1.00E-34         | 1.03E-32        | <b>3.29E-01</b> | 5.12E-01        | 7.96E-05        | 1.68E-68         | 8.79E-72         |
|           | AVG     | 6.33E-70         | 2.23E-36         | 5.84E-35         | 2.28E-32        | 5.62E+01        | 3.10E+00        | 9.02E-05        | 4.75E-69         | 7.59E-72         |
|           | Runtime | 1.3110 s         | 1.8364s          | 1.7205 s         | 1.3215 s        | 1.6028 s        | 1.3710 s        | 1.4187 s        | 1.6062 s         | 1.3945 s         |
| FSA       | Best    | <b>5.00E-256</b> | <b>5.23E-134</b> | <b>7.70E-249</b> | <b>7.64E-89</b> | 1.75E+01        | 1.01E+00        | <b>3.65E-06</b> | <b>2.01E-184</b> | <b>3.46E-261</b> |
|           | STD     | <b>0.00E+00</b>  | <b>2.09E-120</b> | <b>0.00E+00</b>  | <b>1.11E-71</b> | 3.74E-01        | 3.32E-01        | <b>4.93E-05</b> | <b>0.00E+00</b>  | <b>0.00E+00</b>  |
|           | AVG     | <b>5.80E-238</b> | <b>1.32E-120</b> | <b>7.91E-222</b> | <b>2.06E-72</b> | 5.44E+01        | 1.55E+00        | <b>6.84E-05</b> | <b>6.00E-170</b> | <b>5.03E-236</b> |
|           | Runtime | 2.1534 s         | 2.6288 s         | 2.5137 s         | 2.1060 s        | 2.3978 s        | 2.1673 s        | 2.2092 s        | 2.4805 s         | 2.1308 s         |

test collects the results of 30 independent tests and finds their mean, standard deviation, optimal value, and running time.

#### A. SINGLE-PEAK TEST FUNCTION

Table 2 gives the formula, search interval, global optimum and test dimension of the single-peaked test function. This type of test function is characterized by only one extreme value point in the search interval. That is, there is only one global optimum solution and no local solution for the test of this type of function. It is convenient to verify the speed of convergence and the accuracy of the search for the optimum of the algorithm.

By testing the single-dimensional benchmark test functions listed in Table 2, the results are obtained and presented in Table 3.

In the above tests, the results of a randomly selected test are plotted to compare the convergence curves of the

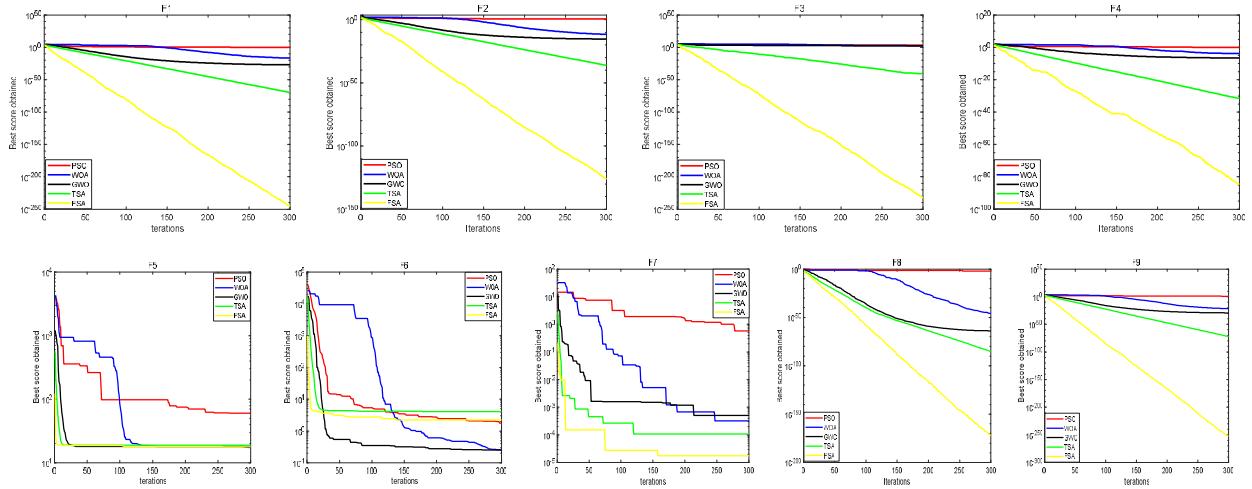
five algorithms. Comparing the convergence performance of the five algorithms is more intuitive, as shown in Fig. 5.

#### 1) CONVERGENCE ACCURACY ANALYSIS

As shown in Table 3, the average value of the optimal solutions found by FSA on the test functions F1–F4 and F8–F9 is better than the results of the other four algorithms. Although FSA does not find an optimal solution for test function F5, its results are significantly better than those of PSO, GWO, and GSA. In addition, although FSA does not have the highest accuracy for test functions F5–F6, its performance is still good among the five algorithms. For test function F7, FSA is slightly better than the other four algorithms in terms of the average and optimal values obtained.

#### 2) ROBUSTNESS ANALYSIS

The test data of standard deviation in Table 3 show that the standard deviation of FSA on test functions



**FIGURE 5.** Comparison of convergence curves of single-peaked functions (F1–F9).

**TABLE 4.** Multi-peak test functions.

| Function formula  | Region of search | Best          | Dimensionality |
|---|------------------|---------------|----------------|
| $F_{10}(x) = \sum_{i=1}^n -x_i \times \sin(\sqrt{ x_i })$   | [-500, 500]      | -418.9829 × n | 20             |
| $F_{11}(x) = \sum_{i=1}^n [x_i^2 - 10 \times \cos(2\pi x_i) + 10]$  | [-5.12, 5.12]    | 0             | 20             |
| $F_{12}(x) = -20 \times \exp(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$   | [-32, 32]        | 0             | 20             |
| $F_{13}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$  | [-600, 600]      | 0             | 20             |
| $F_{14}(x) = \sum_{i=1}^{d-1} (\omega_i - 1)^2 [1 + 10 \sin^2(\pi \omega_i + 1)] + \sin^2(\pi \omega_d) + (\omega_d - 1)^2 \times [1 + \sin^2(2\pi \omega_d)], \omega_i = 1 + \frac{x_i - 1}{4}, i = 1, \dots, d$ | [-10, 10]        | 0             | 20             |
| $F_{15}(x) = 418.9529d - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$   | [-500, 500]      | 0             | 2              |
| $F_{16}(x) = 0.5 + \left[ (\sin \sqrt{\sum_{i=1}^D x_i^2})^2 - 0.5 \right] / [1 + 0.001(\sum_{i=1}^D x_i^2)]^2$   | [-10, 10]        | 0             | 20             |

F1–F4 and F8–F9 is extremely low, even 0. The results of are more stable than those of the other four algorithms.

In test function F5, the difference between FSA and the most stable TSA is slight. When dealing with test function F6, the stability of the FSA is in the middle of the range of the five algorithms. For test function F7, FSA is slightly more robust than the other four algorithms. In summary, the simulations show that the FSA is more robust for single-peaked test functions and has a greater advantage over the other four algorithms.

### 3) CONVERGENCE SPEED ANALYSIS

Fig. 5 shows that compared with the other four algorithms, FSA has an absolute advantage in the convergence speed of test functions F1–F4 and F8–F9. When testing F5 and F6, FSA performs only moderately. The convergence curve of FSA in test function F7 is slightly better than those of the other four algorithms.

In summary, when dealing with single-peaked test functions, FSA can quickly and more consistently find an optimal solution close to the target solution.

### B. MULTI-PEAK TEST FUNCTIONS

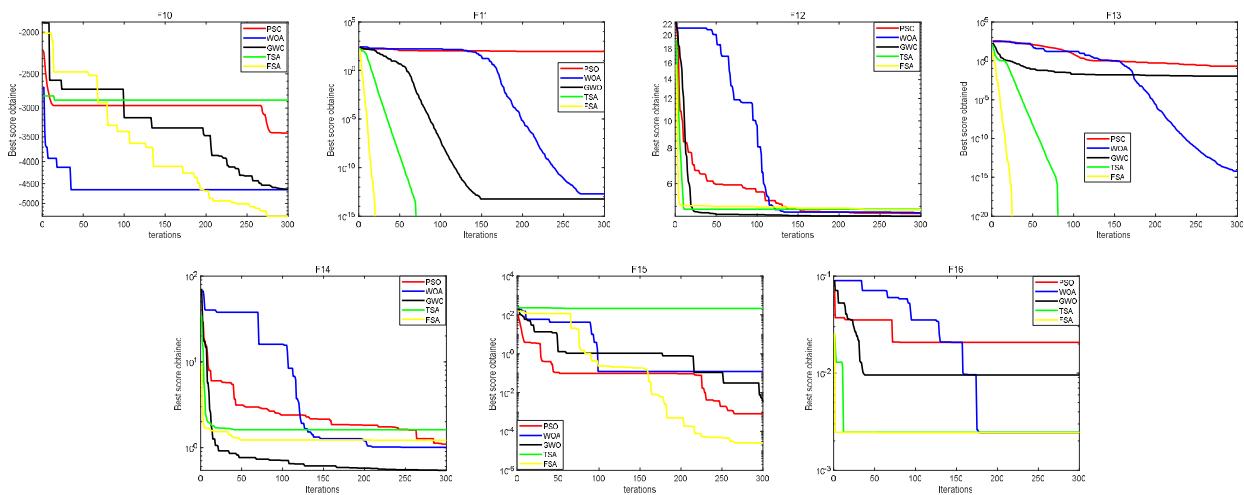
As the multi-peak test function has the characteristic of having multiple local extrema, the algorithm becomes susceptible to the problem of falling into local convergence. The multi-peak test function can therefore be used to test the ability of FSA to escape local extrema and to explore globally. Table 4 gives the formulas, initialization intervals, global optima, and test dimensions for the seven multi-peaked test functions.

The results of testing the multi-peaked test functions listed in Table 4 are shown in Table 5.

Similarly, in the above tests, the results of a randomly selected test are plotted against the convergence curves of the five algorithms as shown in Fig. 6, which provides a more intuitive comparison of the convergence performance

**TABLE 5.** Comparison of multi-peak test functions.

| Algorithm | ATR     | F10              | F11             | F12             | F13             | F14             | F15             | F16             |
|-----------|---------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| PSO       | Best    | -5.00E+03        | 4.63E+01        | <b>4.34E+00</b> | 4.30E-02        | 3.47E-01        | 1.41E-04        | 9.58E-03        |
|           | STD     | 8.22E+02         | 1.87E+01        | 2.26E-01        | 5.73E-02        | 1.49E+00        | 5.43E+01        | 5.61E-03        |
|           | AVG     | -3.60E+03        | 2.65E+02        | 4.79E+00        | 1.85E-01        | 1.83E+00        | 3.55E+01        | 1.52E-02        |
|           | Runtime | <b>1.3016 s</b>  | <b>1.0998 s</b> | <b>1.2312 s</b> | <b>1.2303 s</b> | <b>1.6590 s</b> | 0.2713 s        | <b>1.1585 s</b> |
| WOA       | Best    | -5.90E+03        | <b>0.00E+00</b> | 4.65E+00        | <b>0.00E+00</b> | 4.82E-01        | 6.53E-05        | 7.21E-09        |
|           | STD     | 3.67E+02         | 2.29E-12        | 3.90E-02        | 1.55E-02        | 1.61E-01        | 6.42E-01        | 1.85E-03        |
|           | AVG     | -4.54E+03        | 2.31E-12        | 4.72E+00        | 6.78E-03        | 8.21E-01        | 3.75E-01        | 2.84E-03        |
|           | Runtime | 1.5919 s         | 1.3673 s        | 1.4748 s        | 1.5258 s        | 1.9499 s        | 0.1984 s        | 1.4406 s        |
| GWO       | Best    | -5.76E+03        | 5.68E-14        | 4.61E+00        | <b>0.00E+00</b> | <b>1.96E-01</b> | 9.84E-05        | 2.45E-03        |
|           | STD     | 7.48E+02         | 4.18E+00        | <b>1.54E-02</b> | 7.96E-03        | 1.37E-01        | 5.43E+01        | 3.44E-03        |
|           | AVG     | -4.26E+03        | 1.08E+01        | <b>4.63E+00</b> | 4.04E-03        | <b>4.42E-01</b> | 3.55E+01        | 6.97E-03        |
|           | Runtime | 2.4656 s         | 2.2490 s        | 2.3855 s        | 2.4578 s        | 3.2690 s        | 0.2971 s        | 2.2682 s        |
| TSA       | Best    | -2.92E+03        | <b>0.00E+00</b> | 4.72E+00        | <b>0.00E+00</b> | 9.06E-01        | 1.88E-01        | 3.70E-06        |
|           | STD     | <b>3.62E+02</b>  | <b>0.00E+00</b> | 4.56E-02        | 5.92E-03        | 3.49E-01        | 1.04E+02        | 4.39E-04        |
|           | AVG     | -2.21E+03        | <b>0.00E+00</b> | 4.80E+00        | 1.93E-03        | 1.62E+00        | 1.15E+02        | 2.37E-03        |
|           | Runtime | 1.4486 s         | 1.3244 s        | 1.4228 s        | 1.4353 s        | 1.8562 s        | <b>0.1894 s</b> | 1.3766 s        |
| FSA       | Best    | <b>-6.48E+03</b> | <b>0.00E+00</b> | 4.74E+00        | <b>0.00E+00</b> | 7.15E-01        | <b>2.55E-06</b> | <b>0.00E+00</b> |
|           | STD     | 4.04E+02         | <b>0.00E+00</b> | 4.37E-02        | <b>0.00E+00</b> | <b>8.95E-02</b> | <b>5.47E-05</b> | <b>1.12E-03</b> |
|           | AVG     | <b>-5.05E+05</b> | <b>0.00E+00</b> | 4.86E+00        | <b>0.00E+00</b> | 9.38E-01        | <b>1.36E-05</b> | <b>1.71E-03</b> |
|           | Runtime | 2.2727 s         | 2.0457 s        | 2.2597 s        | 2.2468 s        | 2.6554 s        | 0.2816 s        | 2.1850 s        |

**FIGURE 6.** Comparison of convergence curves of multi-peak test functions (F10–F16).

and global optimization seeking ability of the five algorithms.

### 1) CONVERGENCE ACCURACY ANALYSIS

As can be seen from Table 5, FSA outperforms the other three algorithms when testing F10, F11, F13, F15, and F16. For test function F12, there is not much difference in the performance of these five algorithms in finding the best. The FSA outperforms the TSA and PSO in test function F14. In conclusion, FSA has a strong global exploration capability when solving multi-peaked test functions.

### 2) ROBUSTNESS ANALYSIS

When testing F11, F13, F15, and F16, FSA outperforms the other four algorithms in terms of stability. When testing

F10 and F12, the robustness of FSA performs mediocrely. In conclusion, FSA has better stability and stronger adaptability when dealing with multi-peaked test functions.

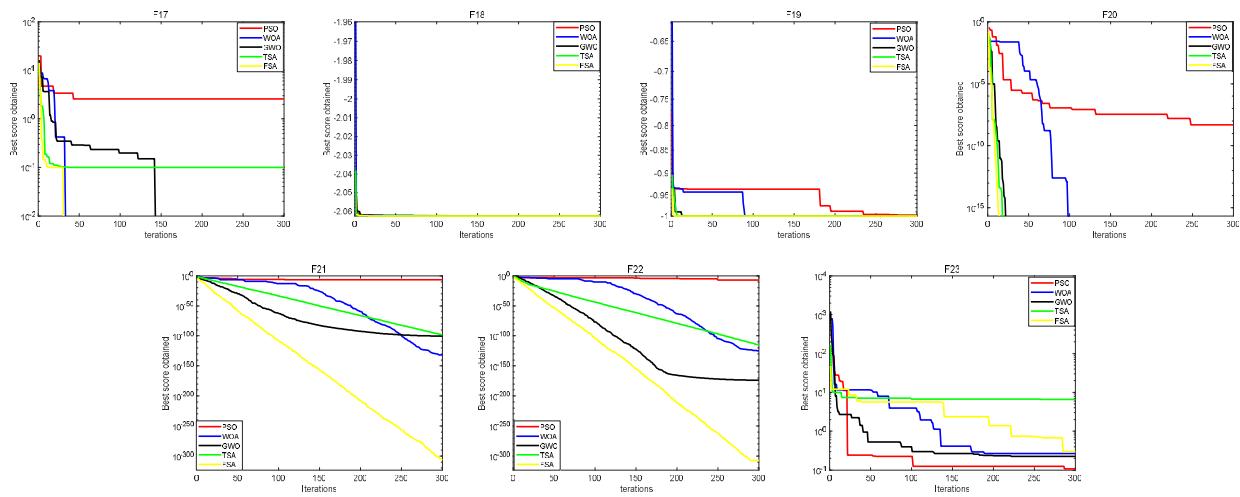
### 3) CONVERGENCE SPEED ANALYSIS

From the convergence curves of F10, F12, and F15 in Fig. 6, it can be reasonably concluded that FSA converges significantly faster than the other four algorithms. For the remaining test functions, FSA also yields very competitive results. As can be seen from the figure, FSA has a faster convergence speed when dealing with high-dimensional and complex problems.

In summary, the treatment of multi-peaked test functions shows that FSA has a strong global search capability and is adaptable to a variety of test functions.

**TABLE 6.** Fixed-dimensional test functions.

| Function formula  | Region of search | Best     | Dimensionality |
|---|------------------|----------|----------------|
| $F_{17}(x) = 100\sqrt{ x_2 - 0.01x_1^2 } + 0.01 x_1 + 10 $  | [-10,1]          | 0        | 2              |
| $F_{18}(x) = -0.0001\left(\left \sin(x_1)\sin(x_2)\exp\left( 100 - \sqrt{x_1^2 + x_2^2} /\pi\right)\right  + 1\right)^{0.1}$                  | [-10,10]         | -2.06261 | 2              |
| $F_{19}(x) = -1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)/0.5(x_1^2 + x_2^2) + 2$   | [-5,12,5,12]     | -1       | 2              |
| $F_{20}(x) = 0.5 + \sin^2(x_1^2 - x_2^2) - 0.5/\left[1 + 0.001(x_1^2 + x_2^2)\right]^2$   | [-100,100]       | 0        | 2              |
| $F_{21}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$  | [-10,10]         | 0        | 2              |
| $F_{22}(x) = 2x_1^2 - 1.05x_1^4 + x_1^6/6 + x_1x_2 + x_2^2$   | [-5,5]           | 0        | 2              |
| $F_{23}(x) = 100(x_1^2 - x_2^2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | [-10,10]         | 0        | 4              |

**FIGURE 7.** Comparison of convergence curves of fixed-dimensional functions (F17–F23).

### C. FIXED-DIMENSIONAL TEST FUNCTIONS

To more comprehensively test the performance of FSA in finding the optimum, several fixed-dimensional test functions are selected to further verify the convergence speed, stability, and convergence accuracy of the algorithm. Table 6 lists the formulas, initialization intervals, global optima, and test dimensions of the fixed-dimension test functions.

The test functions listed in Table 6 above with fixed dimensions are tested and the results obtained are presented in Table 7.

Similarly, in the above tests, the results of a randomly selected test are plotted against the convergence curves of the five algorithms as shown in Fig. 7, which provides a more intuitive comparison of the convergence performance and global optimization seeking ability of the five algorithms.

#### 1) CONVERGENCE ACCURACY ANALYSIS

According to the average test results in Table 7, the convergence accuracy of FSA is significantly higher than those of the other four algorithms when testing F17–F22. The conver-

gence accuracy of FSA is lower than that of WOA but higher than that of PSO, GWO, and TSA when testing F23. This result also proves that these algorithms are very successful in optimizing the fixed-dimensional test functions.

#### 2) STABILITY ANALYSIS

According to Table 7, the stability of the FSA is better than those of the other four algorithms when testing F18–F23. When testing F17, the stability of FSA is slightly lower than that of the WOA but significantly higher than those of the other three algorithms.

#### 3) CONVERGENCE SPEED ANALYSIS

The convergence curve is shown in Fig. 7. When testing F21 and F22, the convergence speed of the FSA is significantly better than the other four algorithms. In testing F17, F18, F19, and F20, the FSA slightly outperforms the other four algorithms. When testing F23, the performance of FSA is average.

The simulation results show that FSA has a strong optimization capability for single-peak test functions, multi-peak

**TABLE 7.** Comparison of fixed-dimensional function tests.

| Algorithm | ATR     | F17             | F18              | F19              | F20             | F21              | F22              | F23             |
|-----------|---------|-----------------|------------------|------------------|-----------------|------------------|------------------|-----------------|
| PSO       | Best    | 1.02E-01        | <b>-2.06E+00</b> | <b>-1.00E+00</b> | 4.12E-12        | 4.61E-09         | 3.54E-08         | 4.09E-02        |
|           | STD     | 9.02E-01        | 7.42E-07         | 2.78E-03         | 6.39E-09        | 2.63E-07         | 7.56E-06         | 1.59E+00        |
|           | AVG     | 1.10E+00        | -2.06E+00        | -9.99E-01        | 3.47E-09        | 2.16E-07         | 3.62E-06         | 9.50E-01        |
|           | Runtime | 0.2638 s        | 0.2567 s         | 0.2665 s         | 0.2652 s        | 0.2507 s         | 0.2633 s         | 0.3936 s        |
| WOA       | Best    | 0.00E+00        | <b>-2.06E+00</b> | <b>-1.00E+00</b> | <b>0.00E+00</b> | 4.33E-142        | 2.10E-143        | 1.48E-04        |
|           | STD     | <b>1.80E-02</b> | 1.18E-08         | <b>0.00E+00</b>  | <b>0.00E+00</b> | 8.52E-116        | 3.75E-115        | 8.63E-01        |
|           | AVG     | 3.33E-03        | -2.06E+00        | <b>-1.00E+00</b> | <b>0.00E+00</b> | 1.59E-116        | 7.34E-116        | <b>4.96E-01</b> |
|           | Runtime | 0.1974 s        | 0.2054 s         | 0.2054 s         | 0.2114 s        | 0.1891 s         | 0.2044 s         | 0.3800 s        |
| GWO       | Best    | 0.00E+00        | <b>-2.06E+00</b> | <b>-1.00E+00</b> | <b>0.00E+00</b> | 4.44E-124        | 7.71E-211        | <b>4.96E-06</b> |
|           | STD     | 5.14E-02        | 8.10E-09         | 1.14E-02         | <b>0.00E+00</b> | 1.17E-94         | 6.23E-158        | 2.15E+00        |
|           | AVG     | 2.77E-02        | -2.06E+00        | -9.98E-01        | <b>0.00E+00</b> | 2.39E-95         | 1.16E-158        | 1.33E+00        |
|           | Runtime | 0.2812 s        | 0.2943 s         | 0.2976 s         | 0.2904 s        | 0.2702 s         | 0.3142 s         | 0.6009 s        |
| TSA       | Best    | 1.00E-01        | -2.06E+00        | <b>-1.00E+00</b> | <b>0.00E+00</b> | 1.94E-110        | 1.02E-166        | 4.67E-01        |
|           | STD     | 4.80E-02        | 1.04E-05         | <b>0.00E+00</b>  | <b>0.00E+00</b> | 1.83E-98         | 1.15E-110        | 3.00E+00        |
|           | AVG     | 1.25E-01        | -2.06E+00        | <b>-1.00E+00</b> | <b>0.00E+00</b> | 4.85E-99         | 2.20E-111        | 5.39E+00        |
|           | Runtime | <b>0.1854 s</b> | <b>0.1899 s</b>  | <b>0.1915 s</b>  | <b>0.1944 s</b> | <b>0.1825 s</b>  | <b>0.2034 s</b>  | <b>0.3654 s</b> |
| FSA       | Best    | <b>0.00E+00</b> | <b>-1.00E+00</b> | <b>-1.00E+00</b> | <b>0.00E+00</b> | <b>0.00E+00</b>  | <b>0.00E+00</b>  | 4.13E-02        |
|           | STD     | 2.49E-02        | <b>8.84E-11</b>  | <b>0.00E+00</b>  | <b>0.00E+00</b> | <b>0.00E+00</b>  | <b>0.00E+00</b>  | <b>6.06E-01</b> |
|           | AVG     | <b>2.67E-03</b> | <b>-2.06E+00</b> | <b>-1.00E+00</b> | <b>0.00E+00</b> | <b>6.08E-301</b> | <b>9.18E-304</b> | 5.41E-01        |
|           | Runtime | 0.2702 s        | 0.2857 s         | 0.2748 s         | 0.2716 s        | 0.2579 s         | 0.2888 s         | 0.5544 s        |

**TABLE 8.** IEEE CEC-2015 benchmark test functions.

| No.    | Functions  | Related basic functions   | Dim | $f_{min}$ |
|--------|--|---|-----|-----------|
| CEC-1  | Rotated Bent Cigar Function  | Bent Cigar Function   | 30  | 100       |
| CEC-2  | Rotated Discus Function  | Discus Function   | 30  | 200       |
| CEC-3  | Shifted and Rotated Weierstrass Function                           | Weierstrass Function  | 30  | 300       |
| CEC-4  | Shifted and Rotated Schwefel's Function                            | Schwefel's Function   | 30  | 400       |
| CEC-5  | Shifted and Rotated Katsuura Function                              | Katsuura Function   | 30  | 500       |
| CEC-6  | Shifted and Rotated HappyCat Function                              | HappyCat Function   | 30  | 600       |
| CEC-7  | Shifted and Rotated HGBat Function                                 | HGBat Function  | 30  | 700       |
| CEC-8  | Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function | Griewank's Function; Rosenbrock's Function  | 30  | 800       |
| CEC-9  | Shifted and Rotated Expanded Scaffer's F6 Function                 | Expanded Scaffer's F6 Function  | 30  | 900       |
| CEC-10 | Hybrid Function 1 ( $N = 3$ )                                      | Schwefel's Function; Rastrigin's Function; High Conditioned Elliptic Function   | 30  | 1000      |
| CEC-11 | Hybrid Function 2 ( $N = 4$ )                                      | Griewank's Function; Weierstrass Function; Rosenbrock's Function; Scaffer's F6 Function   | 30  | 1100      |
| CEC-12 | Hybrid Function 3 ( $N = 5$ )                                      | Katsuura Function; HappyCat Function; Schwefel's Function; Expanded Griewank's plus Rosenbrock's Function; Ackley's Function        | 30  | 1200      |
| CEC-13 | Composition Function 1 ( $N = 5$ )                                 | Rosenbrock's Function; Bent Cigar Function; Discus Function; High Conditioned Elliptic Function; High Conditioned Elliptic Function | 30  | 1300      |
| CEC-14 | Composition Function 2 ( $N = 3$ )                                 | Schwefel's Function; Rastrigin's Function; High Conditioned Elliptic Function   | 30  | 1400      |
| CEC-15 | Composition Function 3 ( $N = 5$ )                                 | HGBat Function; Rastrigin's Function; Schwefel's Function; Weierstrass Function; High Conditioned Elliptic Function                 | 30  | 1500      |

test functions, and fixed-dimensional test functions. In particular, when solving complex fixed-dimensional test functions, FSA has a more obvious competitive advantage over the other three algorithms.

#### D. EVALUATION OF IEEE CEC-2015 TEST FUNCTIONS

Table 8 is the CEC-2015 benchmark function set. Parameter selection of each algorithm in our test is shown in Table 1, the maximum number of iterations is 1000, the dimension size is shown in Table 8, and the population number is set to 50. The test results are shown in Table 10. As can be seen from Table 10, when testing 10 benchmark functions such as CEC-2, CEC-6, CEC-7, CEC-8, CEC-9, CEC-10, CEC-11, CEC-13,

CEC-14, and CEC-15, the test results of FSA are better than those of the other four algorithms. PSO optimizes best when testing functions CEC-1, CEC-4, CEC-5, and CEC-12. WOA optimizes best when testing function CEC-3. Combined with these test results, FSA is undoubtedly the best optimization algorithm.

#### E. EVALUATION OF IEEE CEC-2017 TEST FUNCTIONS

Table 9 shows the benchmark functions for CEC-2017. Parameter selection of each algorithm in our test is shown in Table 1, the maximum number of iterations is 1000, the dimension size is shown in Table 9, and the population number is set to 50. The test results are shown in Table 11.

**TABLE 9.** IEEE CEC-2017 benchmark test functions.

| No.    | Functions   | Dim | $f_{min}$ |
|--------|---|-----|-----------|
| CEC-1  | Shifted and Rotated Bent Cigar Function                 | 30  | 100       |
| CEC-2  | Shifted and Rotated Sum of Different Power Function     | 30  | 200       |
| CEC-3  | Shifted and Rotated Zakharov Function                   | 30  | 300       |
| CEC-4  | Shifted and Rotated Rosenbrock's Function               | 30  | 400       |
| CEC-5  | Shifted and Rotated Rastrigin's Function                | 30  | 500       |
| CEC-6  | Shifted and Rotated Expanded Scaffer's Function         | 30  | 600       |
| CEC-7  | Shifted and Rotated Lunacek Bi_Rastrigin Function       | 30  | 700       |
| CEC-8  | Shifted and Rotated Non-Continuous Rastrigin's Function | 30  | 800       |
| CEC-9  | Shifted and Rotated Levy Function                       | 30  | 900       |
| CEC-10 | Shifted and Rotated Schwefel's Function                 | 30  | 1000      |
| CEC-11 | Hybrid Function 1 ( $N = 3$ )                           | 30  | 1100      |
| CEC-12 | Hybrid Function 2 ( $N = 3$ )                           | 30  | 1200      |
| CEC-13 | Hybrid Function 3 ( $N = 3$ )                           | 30  | 1300      |
| CEC-14 | Hybrid Function 4 ( $N = 4$ )                           | 30  | 1400      |
| CEC-15 | Hybrid Function 5 ( $N = 4$ )                           | 30  | 1500      |
| CEC-16 | Hybrid Function 6 ( $N = 4$ )                           | 30  | 1600      |
| CEC-17 | Hybrid Function 6 ( $N = 5$ )                           | 30  | 1700      |
| CEC-18 | Hybrid Function 6 ( $N = 5$ )                           | 30  | 1800      |
| CEC-19 | Hybrid Function 6 ( $N = 5$ )                           | 30  | 1900      |
| CEC-20 | Hybrid Function 6 ( $N = 6$ )                           | 30  | 2000      |
| CEC-21 | Composition Function 1 ( $N = 3$ )                      | 30  | 2100      |
| CEC-22 | Composition Function 2 ( $N = 3$ )                      | 30  | 2200      |
| CEC-23 | Composition Function 3 ( $N = 4$ )                      | 30  | 2300      |
| CEC-24 | Composition Function 4 ( $N = 4$ )                      | 30  | 2400      |
| CEC-25 | Composition Function 5 ( $N = 5$ )                      | 30  | 2500      |
| CEC-26 | Composition Function 6 ( $N = 5$ )                      | 30  | 2600      |
| CEC-27 | Composition Function 7 ( $N = 6$ )                      | 30  | 2700      |
| CEC-28 | Composition Function 8 ( $N = 6$ )                      | 30  | 2800      |
| CEC-29 | Composition Function 9 ( $N = 3$ )                      | 30  | 2900      |
| CEC-30 | Composition Function 10 ( $N = 3$ )                     | 30  | 3000      |

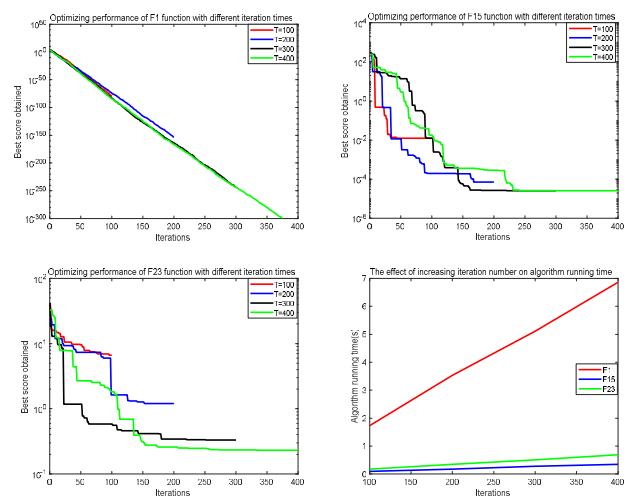
We can see that the test results of 16 of the 30 test functions of FSA are better than those of the other four optimization algorithms. These functions are as follows: CEC-2, CEC-4, CEC-7, CEC-8, CEC-9, CEC-11, CEC-13, CEC-14, CEC-18, CEC-19, CEC-21, CEC-22, CEC-23, CEC-25, CEC-26, and CEC-27. PSO has 12 test functions that perform better than the other four algorithms, these functions are: CEC-1, CEC-3, CEC-5, CEC-6, CEC-10, CEC-12, CEC-15, CEC-16, CEC-17, CEC-20, CEC-24, and CEC-29. The test results of two test functions of GWO are better than those of the other four algorithms, these functions are: CEC-28 and CEC-30. Combined with these test results, FSA is undoubtedly the best optimization algorithm.

#### IV. COMPARISON TEST OF INPUT PARAMETERS

##### A. COMPARISON OF THE NUMBER OF DIFFERENT ITERATIONS

Test function F1, F15, and F23 were chosen to test the variation in the performance of FSA under different numbers of iterations. The number of iterations was 100, 200, 300, and 400. The dimensionality of F1 was fixed at 50 dimensions, the dimensionality of F15 was fixed at 2 dimensions, and the dimensionality of the F23 function was fixed at 4 dimensions. The number of populations was fixed at 50, and the results were shown in Fig. 8.

Fig. 8 shows the relationship between the optimization effect of FSA, the running time and the number of iterations, and the inconsistency in the magnitude of the test variation of

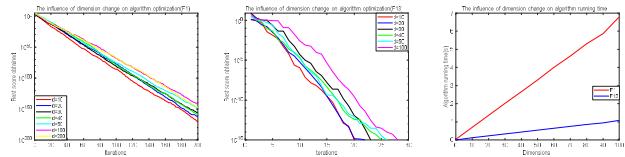


**FIGURE 8.** Comparison of the number of different iterations of FSA (F1, F15, F23).

FSA as it has different standard deviations in optimizing the three different functions mentioned above. The smaller the standard deviation of the algorithm in optimizing a function, the more stable the algorithm is in optimizing that function. The test results in subsection III show that the standard deviation of F1 is 0.00E+00, the standard deviation of F15 is 5.47E-05, and the standard deviation of F23 is 0.60587549. F1 is extremely stable, followed by F15, with F23 as the most chaotic. The above graph demonstrates that as the number of

**TABLE 10.** AVG and STD deviation of best optimal solution for 30 independent runs on CEC-2015 benchmark test functions.

| FUNC   | PSO              |                  | WOA              |           | GWO       |                  | TSA       |           | FSA              |                  |
|--------|------------------|------------------|------------------|-----------|-----------|------------------|-----------|-----------|------------------|------------------|
|        | AVG              | STD              | AVG              | STD       | AVG       | STD              | AVG       | STD       | AVG              | STD              |
| CEC-1  | <b>9.334E+06</b> | 7.223E+06        | 1.071E+08        | 3.998E+07 | 1.198E+09 | 4.136E+08        | 2.588E+08 | 1.649E+08 | 9.721E+06        | <b>7.223E+05</b> |
| CEC-2  | 2.735E+05        | <b>1.444E+06</b> | 6.194E+08        | 3.513E+08 | 7.336E+10 | 8.988E+09        | 2.330E+10 | 7.591E+09 | <b>2.397E+05</b> | 2.546E+06        |
| CEC-3  | 3.209E+02        | 8.625E-02        | <b>3.207E+02</b> | 1.311E-01 | 3.210E+02 | <b>4.919E-02</b> | 3.210E+02 | 5.361E-02 | 3.214E+02        | 8.532E-02        |
| CEC-4  | <b>4.699E+02</b> | <b>2.068E+01</b> | 6.940E+02        | 5.554E+01 | 7.771E+02 | 2.645E+01        | 7.291E+02 | 3.041E+01 | 5.350E+02        | 1.197E+02        |
| CEC-5  | <b>4.856E+03</b> | 1.252E+03        | 5.942E+03        | 6.052E+02 | 7.878E+03 | <b>4.171E+02</b> | 7.106E+03 | 5.964E+02 | 5.779E+03        | 1.454E+03        |
| CEC-6  | 9.822E+05        | 8.333E+05        | 4.387E+06        | 3.638E+06 | 3.856E+07 | 3.663E+07        | 7.359E+06 | 3.583E+06 | <b>7.592E+05</b> | <b>7.293E+04</b> |
| CEC-7  | 7.122E+02        | 2.631E+00        | 7.602E+02        | 3.824E+01 | 1.025E+03 | 9.190E+01        | 8.138E+02 | 4.641E+01 | <b>7.072E+02</b> | <b>1.936E+00</b> |
| CEC-8  | 2.560E+05        | <b>1.407E+05</b> | 8.712E+05        | 4.564E+05 | 1.258E+07 | 8.711E+06        | 2.078E+06 | 1.884E+06 | <b>1.690E+05</b> | 1.603E+05        |
| CEC-9  | 1.026E+03        | 5.681E+01        | 1.076E+03        | 1.504E+02 | 1.381E+03 | 1.254E+02        | 1.130E+03 | 1.160E+02 | <b>1.000E+03</b> | <b>8.238E+00</b> |
| CEC-10 | 3.200E+05        | 3.064E+05        | 5.848E+06        | 2.761E+06 | 3.760E+07 | 3.098E+07        | 1.265E+07 | 1.057E+07 | <b>8.730E+04</b> | <b>4.045E+04</b> |
| CEC-11 | 1.826E+03        | <b>1.190E+02</b> | 2.416E+03        | 2.966E+02 | 2.523E+03 | 4.311E+02        | 2.050E+03 | 4.007E+02 | <b>1.326E+03</b> | 4.653E+03        |
| CEC-12 | <b>1.324E+03</b> | 3.444E+01        | 1.332E+03        | 3.544E+01 | 1.412E+03 | <b>1.778E+01</b> | 1.357E+03 | 2.690E+01 | 1.300E+00        | 2.103E+01        |
| CEC-13 | 1.300E+03        | 2.733E-02        | 1.300E+03        | 1.705E-02 | 2.048E+03 | 4.961E+02        | 1.334E+03 | 3.939E+01 | <b>1.300E+03</b> | <b>1.038E-02</b> |
| CEC-14 | 3.691E+04        | 2.338E+03        | 3.986E+04        | 2.709E+03 | 1.201E+05 | 1.871E+04        | 6.459E+04 | 8.413E+03 | <b>3.021E+04</b> | <b>1.056E+03</b> |
| CEC-15 | 1.600E+03        | <b>1.342E-02</b> | 1.694E+03        | 9.773E+01 | 3.221E+04 | 1.013E+04        | 3.487E+03 | 8.156E+02 | <b>1.600E+03</b> | 2.990E-02        |

**FIGURE 9.** Comparison of different dimensions of FSA (F1, F13).

iterations increases, the algorithm running time will gradually increase, a conclusion that is line with the time complexity analysis.

### B. COMPARISON OF DIFFERENT DIMENSIONS

To test the variation of the FSA algorithm under different iterations, F1 and F13 were selected for testing. The dimensions of the objective functions were chosen as 10, 20, 30, 50, and 100, with the number of iterations fixed at 200 and the number of populations fixed at 50.

Fig. 9 shows the relationship between the performance and running time of FSA and the dimensionality of the objective function, which is more sensitive to the number of dimensions of the objective function, especially for F13. The speed of convergence of F13 varies slightly more than that of F1 due to the change in dimensionality. Furthermore, as the dimensionality of the objective function increases, FSA's ability to find the optimum gradually decreases and the algorithm runtime gradually increases. This conclusion also follows that from the complexity analysis of FSA.

### C. COMPARISON OF THE NUMBERS OF DIFFERENT POPULATIONS

To test the variation pattern of FSA's superiority-seeking performance under different population sizes, F1, F15, and F23 were selected for testing. The numbers of populations were 10, 20, 30, 50, and 100, with the number of iterations fixed to 200 and the dimension was fixed to 20. The results are shown in Fig. 10.

The sensitivity of FSA to population size is high, especially for F15 and F23, where the change in population size causes

a large change in the convergence speed and accuracy of the algorithm. As the population size of the flamingo increases. FSA's ability to find the optimum increases gradually. The algorithm running time also increases incrementally. This conclusion is also drawn from the complexity analysis of FSA.

### D. ANALYSIS OF THE DEGREES OF FREEDOM OF THE GAUSSIAN DISTRIBUTION

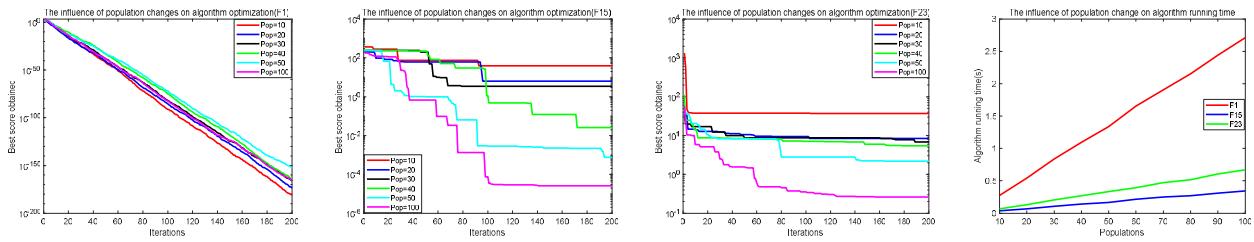
To test and analyze the values of the Gaussian distribution degrees of freedom in updating the formula of FSA, the following experiments are designed in this subsection. Two single-peaked functions F1 and F7, one multi-peaked function F15, and one fixed-dimensional function F23 are selected, with number of iterations fixed at 200 and the number of populations fixed at 50. The dimensionality of F1 and F7 is set to 50, the dimensionality of F15 is set to 2, and the dimensionality of F23 is set to 4.

As can be seen from Fig. 11, the change in the value of the Gaussian distribution degrees of freedom does not affect the running time of the algorithm. Thus, when choosing the degrees of freedom, it is only necessary to consider the effect of the value of that degree of freedom on the algorithm's optimization finding accuracy, stability and convergence speed.

As can be seen from Table 12 and Fig. 12, when testing F1, the smaller the degrees of freedom the better. However, when testing F7, F15, and F23, the standard deviation and mean of FSA search increases as the degrees of freedom increase, reaching a minimum value between 1 and 1.2. Therefore, the Gaussian distribution degrees of freedom are 1–1.2 more appropriate for most of the functions for which FSA performs optimization.

### E. ANALYSIS OF THE VALUE OF THE CHI-SQUARE DISTRIBUTION DEGREES OF FREEDOM

To test and analyze the values of chi-square distribution degree of freedom in updating the formula of FSA, the following experiments are designed in this subsection. Two single-peaked functions F1 and F7, one multi-peaked

**FIGURE 10.** Comparison of numbers of different populations of FSA (F1, F15, F23).**TABLE 11.** AVG and STD deviation of best optimal solution for 30 independent runs on CEC-2017 benchmark test functions.

| FUNC   | PSO              |                  | WOA       |           | GWO              |                  | TSA       |                  | FSA              |                  |
|--------|------------------|------------------|-----------|-----------|------------------|------------------|-----------|------------------|------------------|------------------|
|        | AVG              | STD              | AVG       | STD       | AVG              | STD              | AVG       | STD              | AVG              | STD              |
| CEC-1  | <b>4.051E+03</b> | <b>4.345E+03</b> | 8.143E+07 | 4.714E+07 | 3.830E+10        | 4.960E+09        | 1.229E+10 | 2.366E+09        | 5.699E+03        | 7.682E+03        |
| CEC-2  | 9.821E+03        | 7.757E+05        | 9.445E+05 | 2.284E+07 | 8.565E+08        | 1.243E+08        | 6.755E+09 | 1.416E+09        | <b>8.592E+03</b> | <b>3.959E+05</b> |
| CEC-3  | <b>2.475E+04</b> | 7.696E+03        | 2.180E+05 | 6.544E+04 | 8.052E+04        | 9.024E+03        | 7.088E+04 | <b>5.247E+03</b> | 8.593E+04        | 9.537E+03        |
| CEC-4  | 5.438E+02        | <b>3.370E+01</b> | 5.948E+02 | 4.866E+01 | 9.035E+03        | 1.523E+03        | 1.691E+03 | 6.143E+02        | <b>1.178E+02</b> | 9.770E+01        |
| CEC-5  | <b>5.682E+02</b> | 1.992E+01        | 7.707E+02 | 6.312E+01 | 8.855E+02        | <b>1.340E+01</b> | 8.105E+02 | 2.820E+01        | 6.832E+02        | 3.840E+01        |
| CEC-6  | <b>6.003E+02</b> | 3.125E-01        | 6.733E+02 | 1.354E+01 | 6.869E+02        | 7.153E+00        | 6.774E+02 | 6.240E+00        | 6.433E+02        | <b>1.135E-01</b> |
| CEC-7  | 8.306E+02        | 2.638E+01        | 1.245E+03 | 7.021E+01 | 1.323E+03        | 5.568E+01        | 1.254E+03 | 7.162E+01        | <b>8.086E+02</b> | <b>2.338E+01</b> |
| CEC-8  | 8.701E+02        | 2.085E+01        | 1.015E+03 | 6.093E+01 | 1.123E+03        | <b>1.706E+01</b> | 1.042E+03 | 2.518E+01        | <b>6.721E+02</b> | 3.606E+01        |
| CEC-9  | 1.086E+03        | <b>2.321E+02</b> | 9.401E+03 | 3.258E+03 | 9.463E+03        | 9.912E+02        | 8.858E+03 | 1.010E+03        | <b>1.026E+03</b> | 9.711E+02        |
| CEC-10 | <b>2.016E+03</b> | 1.196E+03        | 6.521E+03 | 9.797E+02 | 8.535E+03        | <b>4.171E+02</b> | 7.470E+03 | 7.910E+02        | 4.901E+03        | 6.811E+02        |
| CEC-11 | 1.229E+03        | <b>6.346E+01</b> | 2.776E+03 | 1.193E+03 | 6.274E+03        | 1.420E+03        | 3.153E+03 | 1.110E+03        | <b>1.163E+03</b> | 5.215E+02        |
| CEC-12 | <b>5.668E+04</b> | <b>1.074E+05</b> | 9.475E+07 | 6.715E+07 | 9.648E+09        | 2.264E+09        | 1.330E+09 | 9.687E+08        | 6.608E+04        | 9.587E+05        |
| CEC-13 | 1.584E+04        | 1.647E+04        | 2.304E+05 | 2.795E+05 | 6.294E+09        | 2.674E+09        | 2.594E+08 | 2.054E+08        | <b>1.464E+04</b> | <b>1.020E+03</b> |
| CEC-14 | 2.482E+04        | <b>2.207E+04</b> | 2.063E+06 | 1.886E+06 | 1.559E+06        | 8.637E+05        | 1.255E+06 | 6.777E+05        | <b>4.542E+04</b> | 6.207E+04        |
| CEC-15 | <b>1.479E+04</b> | <b>2.088E+04</b> | 1.009E+05 | 8.023E+04 | 1.015E+05        | 1.377E+08        | 4.397E+06 | 4.100E+06        | 1.838E+04        | 8.188E+04        |
| CEC-16 | <b>2.429E+03</b> | 2.868E+02        | 3.839E+03 | 4.442E+02 | 4.979E+03        | 4.989E+02        | 3.724E+03 | 4.461E+02        | 2.643E+03        | <b>1.468E+02</b> |
| CEC-17 | <b>1.955E+03</b> | <b>1.376E+02</b> | 2.628E+03 | 2.599E+02 | 3.177E+03        | 3.088E+02        | 2.541E+03 | 2.516E+02        | 2.555E+03        | 5.476E+02        |
| CEC-18 | 1.003E+06        | 1.487E+06        | 3.614E+06 | 3.299E+06 | 1.302E+07        | 1.070E+07        | 6.557E+06 | 3.433E+06        | <b>5.200E+05</b> | <b>9.736E+05</b> |
| CEC-19 | 1.822E+04        | 1.415E+04        | 5.245E+06 | 4.896E+06 | 1.529E+08        | 9.928E+07        | 6.414E+06 | 3.015E+06        | <b>5.224E+03</b> | <b>1.285E+04</b> |
| CEC-20 | <b>2.255E+03</b> | 1.392E+02        | 2.796E+03 | 1.946E+02 | 2.888E+03        | 1.744E+02        | 2.764E+03 | 1.979E+02        | 2.585E+03        | <b>1.072E+02</b> |
| CEC-21 | 2.371E+03        | 3.322E+01        | 2.589E+03 | 5.280E+01 | 2.683E+03        | <b>3.133E+01</b> | 2.611E+03 | 3.865E+01        | <b>2.211E+03</b> | 1.193E+02        |
| CEC-22 | 4.300E+03        | 2.135E+03        | 6.891E+03 | 1.992E+03 | 8.080E+03        | <b>1.145E+03</b> | 8.753E+03 | 1.306E+03        | <b>3.100E+03</b> | 7.245E+03        |
| CEC-23 | 2.719E+03        | <b>2.602E+01</b> | 3.051E+03 | 1.016E+02 | 3.558E+03        | 1.041E+02        | 3.207E+03 | 8.905E+01        | <b>2.591E+03</b> | 1.120E+02        |
| CEC-24 | <b>2.745E+03</b> | <b>4.762E+01</b> | 3.189E+03 | 9.263E+01 | 3.861E+03        | 1.431E+02        | 3.414E+03 | 8.819E+01        | 2.965E+03        | 2.656E+02        |
| CEC-25 | 2.910E+03        | 1.532E+01        | 3.004E+03 | 3.260E+01 | 4.341E+03        | 3.361E+02        | 3.309E+03 | 1.719E+02        | <b>2.783E+03</b> | <b>9.317E+00</b> |
| CEC-26 | 4.236E+03        | 4.054E+02        | 7.549E+03 | 9.121E+02 | 9.773E+03        | <b>3.579E+02</b> | 8.301E+03 | 9.384E+02        | <b>3.136E+03</b> | 8.554E+02        |
| CEC-27 | 3.248E+03        | <b>1.660E+01</b> | 3.398E+03 | 8.753E+01 | 4.471E+03        | 2.400E+02        | 3.699E+03 | 1.558E+02        | <b>2.948E+03</b> | 9.860E+01        |
| CEC-28 | 3.263E+03        | <b>2.734E+01</b> | 3.361E+03 | 3.629E+01 | <b>3.192E+03</b> | 4.390E+02        | 3.998E+03 | 3.526E+02        | 3.357E+03        | 5.134E+01        |
| CEC-29 | <b>3.207E+03</b> | 1.271E+02        | 4.868E+03 | 4.019E+02 | 6.556E+03        | 5.767E+02        | 5.322E+03 | 4.205E+02        | 3.607E+03        | <b>3.514E+01</b> |
| CEC-30 | 3.909E+03        | 4.204E+03        | 9.449E+03 | 6.381E+02 | <b>3.376E+03</b> | 8.689E+02        | 4.784E+03 | <b>2.134E+02</b> | 3.397E+03        | 4.134E+02        |

**TABLE 12.** Comparison of the effect of different values of Gaussian distribution degrees of freedom on the accuracy and stability of the algorithm for finding the optimum (average of 30 iterations).

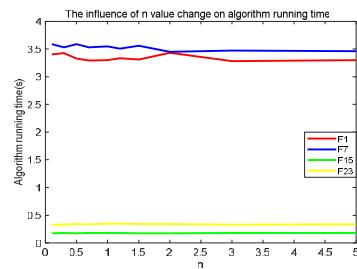
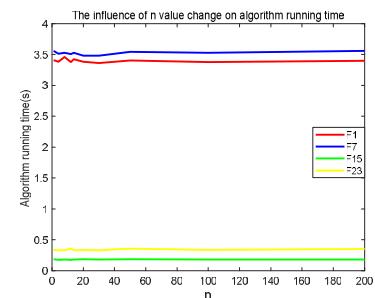
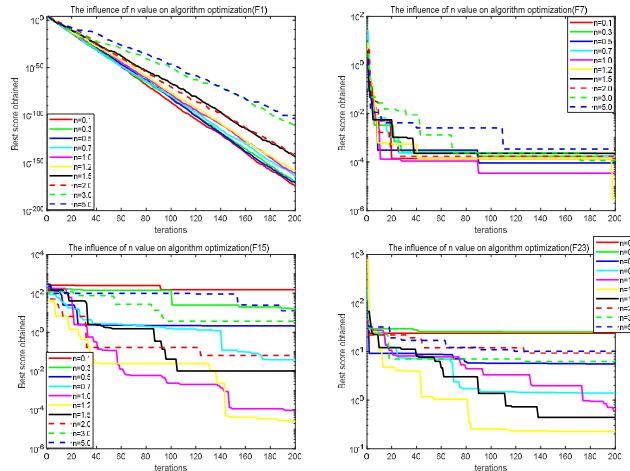
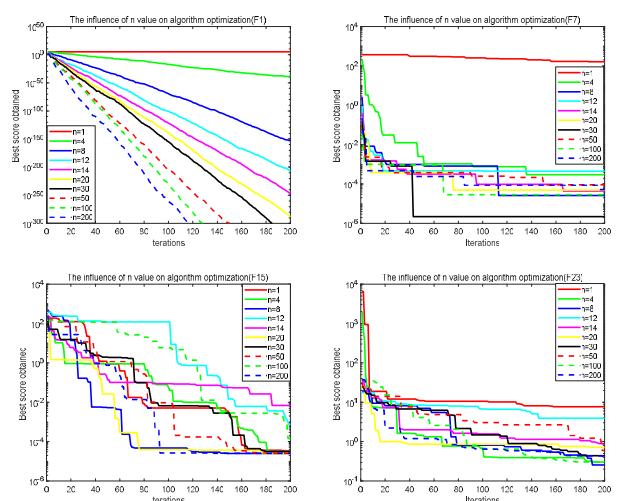
| n   | F1              |                  | F7              |                 | F15             |                 | F23             |                 |
|-----|-----------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|     | STD             | AVG              | STD             | AVG             | STD             | AVG             | STD             | AVG             |
| 0.1 | <b>0.00E+00</b> | <b>1.46E-165</b> | 1.06E-04        | 8.94E-05        | 8.44E+01        | 1.36E+02        | 7.62E+00        | 2.12E+01        |
| 0.3 | 1.73E-160       | 3.25E-161        | 1.41E-04        | 1.08E-04        | 5.08E+01        | 5.85E+01        | 9.18E+00        | 1.64E+01        |
| 0.5 | 4.19E-160       | 7.86E-161        | 1.14E-04        | 1.11E-04        | 3.47E+01        | 1.12E+01        | 5.92E+00        | 6.74E+00        |
| 0.7 | 1.58E-159       | 4.65E-160        | 1.37E-04        | 1.28E-04        | 2.13E+01        | 4.30E+00        | 2.43E+00        | 2.76E+00        |
| 1   | 7.94E-151       | 1.47E-151        | 7.11E-05        | 1.05E-04        | 4.17E-03        | 8.05E-04        | <b>1.04E+00</b> | <b>7.80E-01</b> |
| 1.2 | 1.96E-146       | 4.95E-147        | <b>7.07E-05</b> | <b>7.84E-05</b> | <b>9.58E-05</b> | <b>6.40E-05</b> | 1.78E+00        | 1.35E+00        |
| 1.5 | 7.61E-139       | 1.60E-139        | 1.22E-04        | 1.30E-04        | 3.91E-02        | 1.51E-02        | 2.79E+00        | 2.74E+00        |
| 2   | 4.38E-125       | 8.13E-126        | 1.77E-04        | 1.91E-04        | 2.15E+01        | 4.42E+00        | 3.19E+00        | 3.41E+00        |
| 3   | 7.56E-114       | 1.77E-114        | 1.27E-04        | 1.68E-04        | 1.17E+01        | 6.98E+00        | 3.81E+00        | 5.69E+00        |
| 5   | 7.90E-89        | 1.62E-89         | 3.18E-04        | 3.58E-04        | 4.31E+01        | 4.06E+01        | 3.02E+00        | 5.19E+00        |

function F15, and one fixed-dimensional function F23 are selected, with maximum number of iterations fixed at 200 and the number of populations fixed at 50. The dimensionality

of F1 and F7 is set to 50 dimensions, the dimensionality of F15 is set to 2 dimensions, and the dimensionality of F23 is set to 4 dimensions.

**TABLE 13.** Comparison of the effect of different values of chi-square distribution degrees of freedom on the accuracy and stability of FSA search (average of 30 iterations).

| n   | F1              |                 | F7              |                 | F15             |                 | F23             |                 |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|     | STD             | AVG             | STD             | AVG             | STD             | AVG             | STD             | AVG             |
| 1   | 9.82E+03        | 8.43E+04        | 4.85E+01        | 1.96E+02        | 4.27E-03        | 5.67E-03        | 3.99E+00        | 5.39E+00        |
| 4   | 3.14E-36        | 7.81E-37        | 5.33E-04        | 7.98E-04        | 1.80E-02        | 3.88E-03        | 1.83E+00        | 1.61E+00        |
| 8   | 3.20E-146       | 5.96E-147       | 9.21E-05        | 1.00E-04        | 1.34E-01        | 2.51E-02        | <b>1.17E+00</b> | <b>8.92E-01</b> |
| 12  | <b>0.00E+00</b> | 8.94E-204       | 7.45E-05        | 1.00E-04        | <b>1.15E-03</b> | <b>9.25E-04</b> | 1.87E+00        | 1.97E+00        |
| 14  | <b>0.00E+00</b> | 4.14E-228       | <b>6.74E-05</b> | <b>8.83E-05</b> | 8.27E-02        | 1.69E-02        | 1.19E+00        | 9.46E-01        |
| 20  | <b>0.00E+00</b> | 2.07E-272       | 8.94E-05        | 9.55E-05        | 6.17E-02        | 1.23E-02        | 1.77E+00        | 1.17E+00        |
| 30  | <b>0.00E+00</b> | 2.14e-317       | 6.87E-05        | 8.99E-05        | 1.52E-03        | 1.41E-03        | 1.53E+00        | 9.83E-01        |
| 50  | <b>0.00E+00</b> | <b>0.00E+00</b> | 8.56E-05        | 9.03E-05        | 5.08E-03        | 1.47E-03        | 1.23E+00        | 1.08E+00        |
| 100 | <b>0.00E+00</b> | <b>0.00E+00</b> | 1.02E-04        | 1.14E-04        | 1.57E-03        | 1.03E-03        | 1.74E+00        | 1.26E+00        |
| 200 | <b>0.00E+00</b> | <b>0.00E+00</b> | 8.06E-05        | 8.08E-05        | 5.36E-03        | 1.15E-03        | 2.05E+00        | 1.12E+00        |

**FIGURE 11.** Effect of different values of the Gaussian distribution degrees of freedom on the running time of the algorithm.**FIGURE 13.** Effect of different values of the chi-square distribution degrees of freedom on the running time of the algorithm.**FIGURE 12.** Comparison of the convergence speed of the four functions for different values of the Gaussian distribution degrees of freedom.**FIGURE 14.** Plot of the speed of convergence of the four functions for different values of degrees of freedom in the chi-square distribution.

As can be seen from Fig. 13, the different values of the chi-square distribution degrees of freedom do not affect the running time of the algorithm. Therefore, when choosing the degrees of freedom, it is only necessary to consider the effect of the size of this degree of freedom on the algorithm's optimization accuracy, stability, and convergence accuracy.

As can be seen from Table 13 and Fig. 14, when testing F1, the greater the degrees of freedom, the better. However, when testing F7, F15, and F23, the standard deviation and mean of FSA search increases with the degrees of freedom.

The minimum degrees of freedom taken is between 8 and 14. Therefore, the chi-square distribution degrees of freedom are 8–14 more appropriate for most of the functions tested by FSA.

## V. FSA FOR FINDING THE OPTIMAL TRAJECTORY MAP

To see the population search trajectory of FSA algorithm, three single-peaked functions (F1, F2, F3), three multi-peaked functions (F11, F12, F15) and three

**TABLE 14.** Range of OTL parameters and their physical significance.

| Parameter range          | Physical significance |
|--------------------------|-----------------------|
| $R_{b1} \in [50, 150]$   | Resistor $b_1$        |
| $R_{b2} \in [25, 70]$    | Resistance $b_2$      |
| $R_f \in [0.5, 3]$       | Resistance $f$        |
| $R_{c1} \in [1.2, 2.5]$  | Resistor $c_1$        |
| $R_{c2} \in [0.25, 1.2]$ | Resistor $c_2$        |
| $\beta \in [50, 300]$    | Current increment     |

fixed-dimensional test functions (F19, F21, F22) are selected. Setting the weight of migrating flamingos in the first stage to 10%, the Gaussian distribution degrees of freedom to 8, the chi-square distribution degrees of freedom to 1.2, and the dimensionality to 2, the optimal trajectory of the FSA can be obtained as shown in Fig. 15–17.

The first graph is the three-dimensional stereogram of the original function, the second and third graphs are the optimal trajectory diagram, the fourth graph is the convergence curve diagram, and the fifth graph is the position of each flamingo population at the end of the last iteration.

As can be seen in Fig. 15–17, the flamingo population can move to the optimal point in FSA when seeking different functions, eventually clustering around the global optimum. This phenomenon can be observed when optimizing single-peak, multi-peak, and fixed-dimensional test functions.

## VI. OPTIMIZATION TESTS FOR ENGINEERING APPLICATIONS

### A. TEST SIMULATION IN OUTPUT TRANSFORMER LESS

The relationship between the midpoint voltage and the other parameters in the output transformer less (OTL) is shown in the following equation5.

$$V_m = \frac{(V_{b1} + 0.74) \times \beta \times (R_{c2} + R_l)}{\beta \times (R_{c2} + R_l) + R_f} + \frac{11.35 \times R_f}{\beta \times (R_{c2} + R_l) + R_f} + \frac{0.74 \times R_f \times \beta \times (R_{c2} + 9)}{(\beta \times (R_{c2} + 9) + R_f) \times R_{c1}}, \quad (4)$$

where  $V_{b1} = \frac{(E_c \times R_{b2})}{(R_{b1} + R_{b2})}$ ,

then the output value  $V_m$  of (4) is the midpoint voltage value. The range of values of the parameters and their physical significance are listed in Table 14.

In OTL, the optimal midpoint voltage  $V_m = 1/2 \times E_c$ [26], then the objective function constructed in this subsection is given in (5).

$$\text{Min : } f(x) = |V_m - 1/2 \times E_c| \quad (5)$$

The ideal optimal value in (5) is 0. This sub-section tests the use of FSA to find the optimal parameters by setting  $R_l = 9\Omega$ ,  $E_c = 12V$ . Initializing the number of populations to 20, the maximum number of iterations to 500, and the dimensionality to 6 dimensions, the optimal search curve can be obtained as follows:

Fig. 19 shows that FSA outperforms the other four algorithms in finding the optimal OTL parameters. The values of

**TABLE 15.** Optimal solutions for simulation tests of five algorithms.

| Best     | PSO      | WOA      | GWO      | TSA     | FSA   |
|----------|----------|----------|----------|---------|-------|
| $R_{b1}$ | 85.3056  | 123.8516 | 102.2724 | 55.3447 | 150   |
| $R_{b2}$ | 47.1484  | 63.268   | 49.258   | 37.0291 | 70    |
| $R_f$    | 2.1919   | 1.0967   | 1.4414   | 0.5     | 0.774 |
| $R_{c1}$ | 0.6075   | 1.4768   | 1.26816  | 1.2     | 2.5   |
| $R_{c2}$ | 5.392    | 0.5857   | 1.1272   | 0.3907  | 0.25  |
| $\beta$  | 250.2085 | 151.3161 | 117.5431 | 50      | 50    |

the parameters corresponding to their search for the optimum are shown in Table 15.

### B. APPLICATION OF THE FSA ALGORITHM TO PATH PLANNING

When applied to practical path planning, each flamingo is a solution to the path planning problem. In the path optimization of a mobile robot, each flamingo represents a path of movement for the robot [27], [28]. FSA finds an optimal path among many paths through optimization calculations.

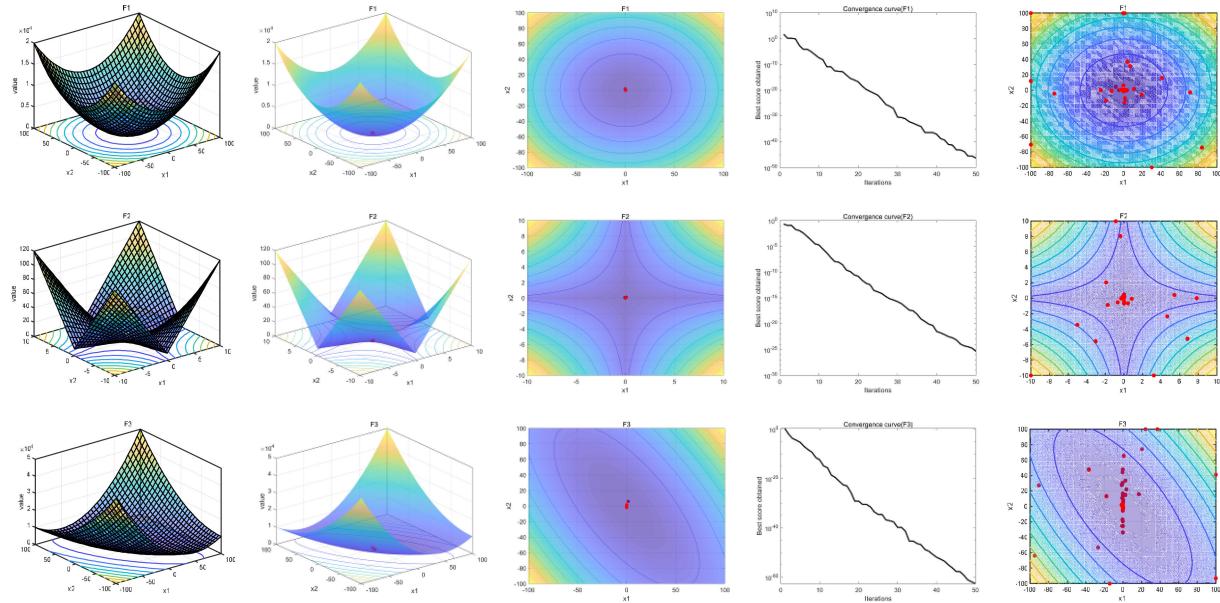
When the flamingo position must be set to correspond to the robot motion model, the result of modelling the robot's motion environment using the raster method is  $m \times n$ , a rectangular area with coordinates starting from 1. The origin raster represents the initial position of the robot, and the raster corresponding to the coordinates  $(m, n)$  represents the moving target position of the robot. An important element of the flamingo position setting is to determine the mathematical representation of the flamingo. This subsection translates this problem into representing the robot's moving path in terms of a vector. Through analysis, the motion path of the robot is found to be continuous in nature, although the model established by the raster method discretizes the space [29], [30].

The robot path optimization model established in this section is shown as follows.

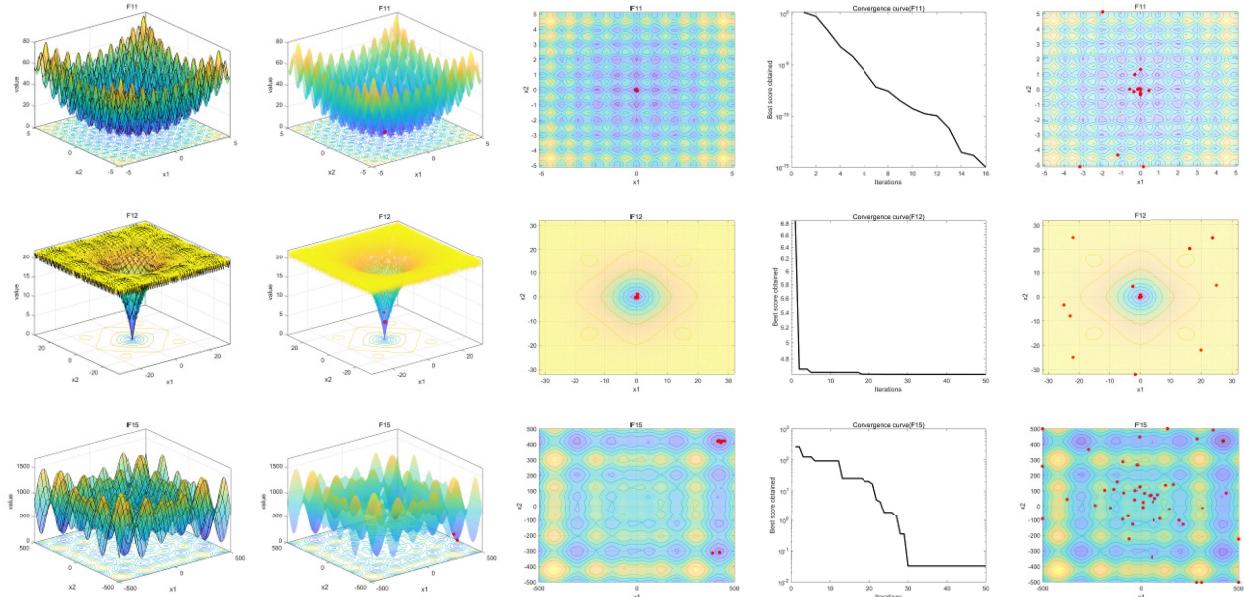
Assuming that the robot workspace is a two-dimensional plane, any point on the plane can be represented as  $(x, y)$ . By dividing the line between the starting point and the end point into  $n$  segments on the Y-axis, drawing  $n$  bisectors parallel to the X-axis, namely,  $\{l_0, l_1, \dots, l_n\}$ , and finding a value of x-coordinate on each bisector, the global path of the robot can be expressed as  $\{ph_0, ph_1, ph_2, \dots, ph_n\}$ . In this paper, the environment is modeled by dividing the line between the start and end points on the Y-axis into  $n$  equal parts, drawing  $n$  lines parallel to the X-axis, and finding a value for each equal line, so that the global path of the robot can be represented as  $\{ph_0, ph_1, ph_2, \dots, ph_n\}$ , where  $ph_0$  is the start point and  $ph_n$  is the target point. As long as the path is  $n$  large enough, the planned path is a continuous curve. Thus, the global path planning problem can be further transformed into a search for the following set of sequential points.

$$\begin{aligned} Path &= \{ph_0, ph_1, ph_2, \dots, ph_n\} \\ &= \{(x_{l1}, y_{l1}), (x_{l2}, y_{l2}), \dots, (x_{ln}, y_{ln})\} \end{aligned} \quad (6)$$

where  $ph_i = (x_{li}, y_{li})$ .



**FIGURE 15.** Optimization-seeking motion trajectory of FSA on F1, F2, and F3.



**FIGURE 16.** Optimization-seeking motion trajectory of FSA on F11, F12, and F15.

The goal of path planning in this section is to have the shortest path length. Based on the modeling approach above, the formula for minimizing the path length is shown in (7).

$$\text{Min} : L(\text{Path}) = \sum_{i=0}^{n-1} \sqrt{(x_{l+1} - x_l)^2 + (y_{l+1} - y_l)^2}, \\ d = (y_{l+1} - y_l) = 1. \quad (7)$$

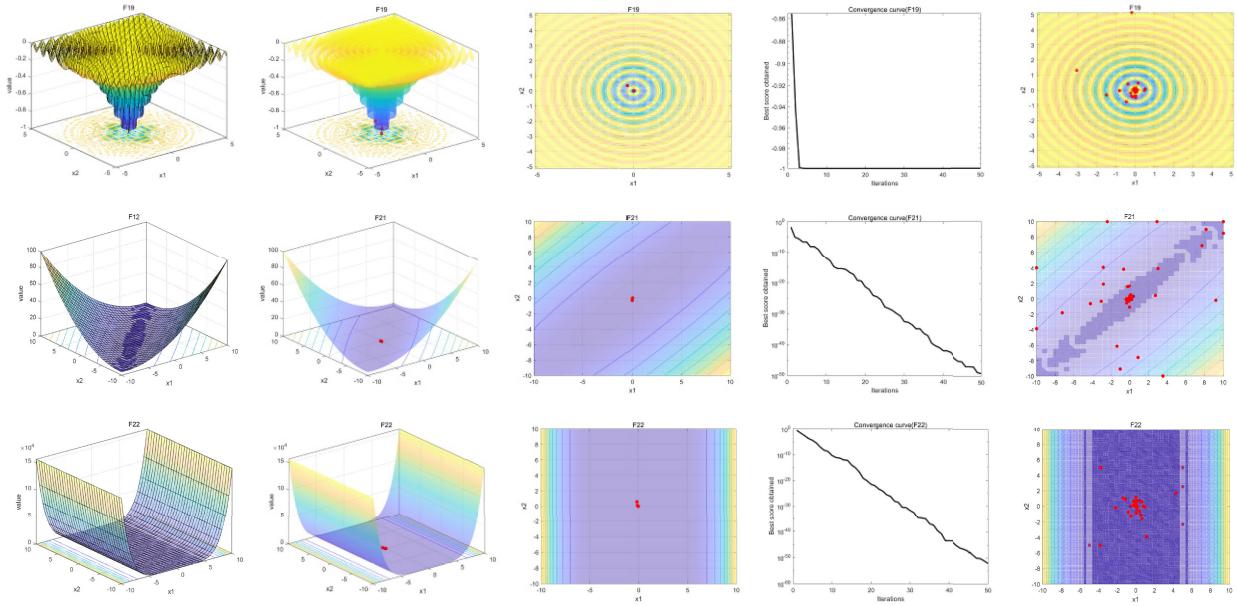
The motion path of the robot is determined by  $(x_1, x_2, \dots, x_n)$ .

The parameters of FSA during the test are set as follows: the number of populations is 50, the maximum number of iterations is 500, the dimension is equal to the number of

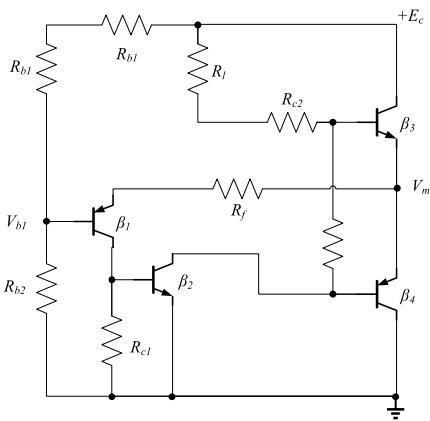
map rows, the lower boundary of the variable is 1, and the upper boundary of the variable is the number of map rows. After the experimental test, the robot motion trajectory and convergence curve results of FSA are shown in Fig. 20.

### C. APPLICATION OF FSA ALGORITHM IN NETWORK INTRUSION DETECTION

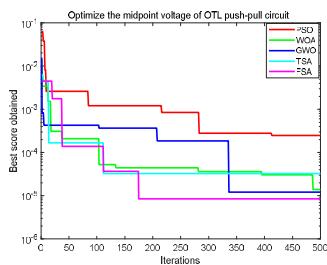
Swarm intelligence optimization algorithms have been used in engineering prediction and evaluation mainly for back propagation of neural networks and parameter optimization of support vector machine (SVM) [31], [32]. In this subsection, SVM model, which is commonly used in network



**FIGURE 17.** Optimization-seeking motion trajectory of FSA on F19, F21, and F22.

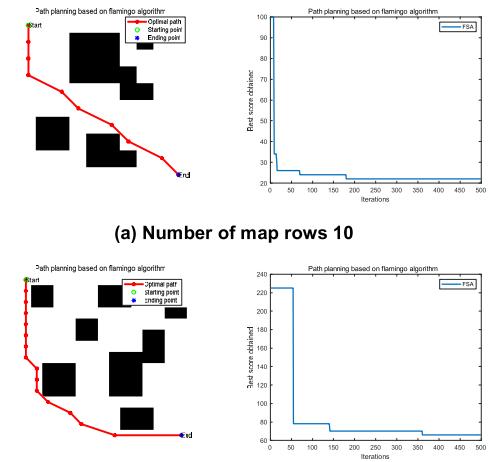


**FIGURE 18.** Circuit diagram of OTL.



**FIGURE 19.** Comparison of OTL simulation test convergence curves.

intrusion detection, is chosen to solve the parameter optimization problem using FSA. FSA is used to find the penalty factor  $C$  in the SVM model and the parameter  $g$  in the radial basis kernel function.

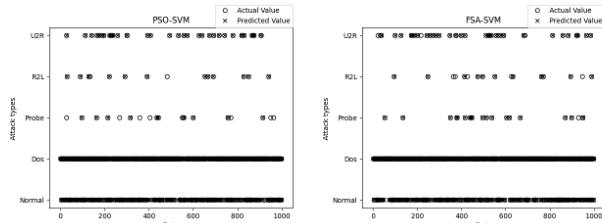


**FIGURE 20.** FSA applied to a path planning problem.

For testing purposes, 5000 randomly selected data items from the KDD99 dataset were used as the data source and were quantified as well as standardized. There are four main types of network attacks in this data set: DoS attacks, Probe attacks, R2L attacks, and U2R attacks [33]. Five sub-datasets were formed by randomly selecting 20% of the processed 5000 data as a sub-dataset. To avoid the influence of other uncontrollable factors on the training accuracy, a ten-fold cross-validation was performed and averaged to compare the prediction results of FSA-SVM and PSO-SVM. The parameters of the two algorithms in the test are set as: lower search limit of 0.001, upper search limit of 100, maximum number of iterations of 20, and dimension of 2. The test results are shown in Table 16. Other parameter settings of the two algorithms are in Table 1.

**TABLE 16.** Comparison of the test results of PSO-SVM and FSA-SVM.

| Models  | Test set Accuracy | Dos prediction accuracy | Probe prediction accuracy | R2L prediction accuracy | U2R prediction accuracy |
|---------|-------------------|-------------------------|---------------------------|-------------------------|-------------------------|
| PSO-SVM | 87.03%            | 84.91%                  | 47.05%                    | 63.15%                  | 81.23%                  |
| FSA-SVM | 92.13%            | 95.25%                  | 90.31%                    | 71.68%                  | 82.26%                  |

**FIGURE 21.** Comparison of PSO-SVM and FSA-SVM test results.

As shown in Table 12, the overall prediction accuracy of the FSA-SVM model is approximately 5% higher than that of the PSO-SVM model, and the prediction accuracy of different types of attacks is improved to varying degrees.

From the 10 randomly selected experimental results, the scatter diagram of the prediction results of the SVM model and the actual results is shown in Fig. 21. Among them, the optimal solution obtained by PSO is  $C = 0.6$ ,  $g = 0.7$ , and the optimal solution obtained by FSA is  $C = 0.3$ ,  $g = 0.005$ . Figure 21 reveals that the FSA-SVM model has a higher prediction accuracy than the PSO-SVM model.

## VII. CONCLUSION

To address the migratory and foraging behavior of flamingos, this paper develops relevant mathematical models that enable FSA to satisfy the global exploration and local exploitation capabilities required from an optimization algorithm. Three sets of experiments are designed to test FSA on 68 test functions, which are used to verify the convergence speed and optimization-seeking accuracy of the algorithm as well as its ability to escape from local extremes and global search. The effect of different input parameters on the FSA is discussed in this paper, and the optimal parameter selection interval is derived. Finally, the simulation results show that the algorithm is highly competitive with existing algorithms in terms of search accuracy, convergence speed, and stability. In addition, nine test functions are selected, and their dimensionality is set to 2. FSA is used to find and visualize their optimization, thus visualizing the movement trajectory of the flamingo population.

To verify the practicality of the proposed algorithm and its optimization performance, practical engineering applications are carried out in this paper, and FSA performs well in three different specific projects. Ultimately, FSA clearly has certain advantages and application value. This new swarm intelligence optimization algorithm can fully provide a new

solution for optimal design in different research areas of the optimization problem. This new solution can better help to solve these engineering problems.

## REFERENCES

- [1] J. Xue, "Research and application of a novel swarm intelligence optimization technique: Sparrow search algorithm," M.S thesis, Donghua Univ., Shanghai, China, 2020.
- [2] C. Liu, T. Wu, and H. Yang, "An algorithm of the outer point fastest drop for the parameter of nonlinear trend model," *Sci. Surveying Mapping*, vol. 44, no. 1, pp. 16–20, Jan. 2018, doi: [10.16251/j.cnki.1009-2307.2019.01.004](https://doi.org/10.16251/j.cnki.1009-2307.2019.01.004).
- [3] X. Liu and C. Chen, "Nested constraint-preferred optimization method for fixed boundary optimal control problem," *J. Zhejiang Univ.*, vol. 44, no. 7, pp. 1247–1250, Jul. 2010, doi: [10.3785/j.issn.1008-973X.2010.07.002](https://doi.org/10.3785/j.issn.1008-973X.2010.07.002).
- [4] X. Li, S. Liu, and W. Xie, "A novel conjugate gradient method for sensing matrix optimization for compressed sensing systems," *J. Zhejiang Univ.*, vol. 46, no. 1, pp. 15–21, Jan. 2019, doi: [10.3785/j.issn.1008-9497.2019.01.003](https://doi.org/10.3785/j.issn.1008-9497.2019.01.003).
- [5] J. Feng and N. Shu, "Applications and theory of swarm intelligence," *Comput. Eng. Appl.*, no. 17, pp. 31–34, Jun. 2002.
- [6] M. G. Hinckey, R. Sterritt, and C. Rouff, "Swarms and swarm intelligence," *Computer*, vol. 40, no. 4, pp. 111–113, Apr. 2007, doi: [10.1109/MC.2007.144](https://doi.org/10.1109/MC.2007.144).
- [7] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013, doi: [10.1109/TCYB.2012.2231860](https://doi.org/10.1109/TCYB.2012.2231860).
- [8] W. Liu, H. Chen, L. Chen, and Y. Chen, "A novel ant colony optimization based algorithm for identifying gene regulatory elements," *J. Comput.*, vol. 8, no. 3, pp. 613–621, Mar. 2013, doi: [10.4304/jcp.8.3.613-621](https://doi.org/10.4304/jcp.8.3.613-621).
- [9] Y. Sun and Q. Xia, "Analysis and improvement of particle group algorithm under semi-supervised clustering target," *J. Beijing Univ. Posts Telecommun.*, vol. 43, no. 5, pp. 21–26, Oct. 2020, doi: [10.13190/j.bjupt.2020-017](https://doi.org/10.13190/j.bjupt.2020-017).
- [10] Y. Zhang, "Coverage optimization and simulation of wireless sensor networks based on particle swarm optimization," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 2, pp. 307–316, Jun. 2020, doi: [10.1007/s10776-019-00446-7](https://doi.org/10.1007/s10776-019-00446-7).
- [11] Y. Lai and J. Zhang, "The optimization of urban bus dispatch based on the analog annealing-adaptive cuckoo algorithm is studied," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 21, no. 1, pp. 183–189, Feb. 2021, doi: [10.16097/j.cnki.1009-6744.2021.01.028](https://doi.org/10.16097/j.cnki.1009-6744.2021.01.028).
- [12] M. S. Othman, S. R. Kumaran, and L. M. Yusuf, "Gene selection using hybrid multi-objective cuckoo search algorithm with evolutionary operators for cancer microarray data," *IEEE Access*, vol. 8, pp. 186348–186361, 2020, doi: [10.1109/ACCESS.2020.3029890](https://doi.org/10.1109/ACCESS.2020.3029890).
- [13] P. Wang, Y. Rao, and Q. Luo, "An effective discrete grey wolf optimization algorithm for solving the packing problem," *IEEE Access*, vol. 8, pp. 115559–115571, 2020, doi: [10.1109/ACCESS.2020.3004380](https://doi.org/10.1109/ACCESS.2020.3004380).
- [14] S. Deng, X. Wang, Y. Zhu, F. Lv, and J. Wang, "Hybrid grey wolf optimization algorithm-based support vector machine for groutability prediction of fractured rock mass," *J. Comput. Civil Eng.*, vol. 33, no. 2, Mar. 2019, Art. no. 04018065, doi: [10.1061/\(ASCE\)CP.1943-5487.0000814](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000814).
- [15] Q. Zhang, Y. Guo, Y. Wang, and X. Liu, "A discrete whale optimization algorithm and application," *J. Univ. Electron. Sci. Technol. China*, vol. 49, no. 4, pp. 622–630, Jul. 2020, doi: [10.12178/1001-0548.2019116](https://doi.org/10.12178/1001-0548.2019116).
- [16] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541, doi: [10.1016/j.engappai.2020.103541](https://doi.org/10.1016/j.engappai.2020.103541).
- [17] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020, doi: [10.1080/21642583.2019.1708830](https://doi.org/10.1080/21642583.2019.1708830).
- [18] J. Zhang, L. Lai, S. Chen, and Y. Fang, "Multi-target optimization of pump turbines based on improved particle group algorithms," *J. Huazhong Univ. Sci. Technol.*, vol. 49, no. 3, pp. 86–92, Feb. 2021, doi: [10.13245/j.hust.210316](https://doi.org/10.13245/j.hust.210316).
- [19] D. Zhang, W. Chen, H. Zhang, and Y. Su, "A algorithm combined with ant colony algorithm for unmanned boat patrol path planning," *J. Huazhong Univ. Sci. Technol.*, vol. 48, no. 6, pp. 13–18, Jun. 2020, doi: [10.13245/j.hust.200603](https://doi.org/10.13245/j.hust.200603).

- [20] X. Lv, X. Mu, J. Zhang, and Z. Wang, "Chaos sparrow search optimization algorithm," *J. Beijing Univ. Aeronaut. Astronaut.*, Aug. 2020, doi: [10.13700/j.bh.1001-5965.2020.0298](https://doi.org/10.13700/j.bh.1001-5965.2020.0298).
- [21] A. Enrique and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 126–142, Apr. 2005.
- [22] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [23] A. Béchet, M. Rendón-martos, M. Á. Rendón, J. A. Amat, A. R. Johnson, and M. Gauthier-Clerc, "Global economy interacts with climate change to jeopardize species conservation: The case of the greater flamingo in the Mediterranean and West Africa," *Environ. Conservation*, vol. 39, no. 1, pp. 1–3, Mar. 2012, doi: [10.1017/S0376892911000488](https://doi.org/10.1017/S0376892911000488).
- [24] F. Ayache, A. M. Gammar, and M. Chaouach, "Environmental dynamics and conservation of the flamingo in the vicinity of Greater Tunis, Tunisia: The case study of Sebkha Essijoumi," *Earth Surf. Processes Landforms*, vol. 31, no. 13, pp. 1674–1684, Nov. 2006, doi: [10.1002/esp.1438](https://doi.org/10.1002/esp.1438).
- [25] W. Xin, "On the calculation of complementary symmetric power amplifier," *Adv. Eng. Sci.*, no. 4, pp. 123–132, 1983, doi: [10.15961/j.jseuse.1983.04.019](https://doi.org/10.15961/j.jseuse.1983.04.019).
- [26] G. Chen, L. Zhang, J. Jian, and D. Wang, "Three designs of voltage stability in the midpoint of the OTL circuit," *J. Appl. Statist. Manage.*, no. 3, pp. 16–27, 1983, doi: [10.13860/j.cnki.sltj.1983.03.004](https://doi.org/10.13860/j.cnki.sltj.1983.03.004).
- [27] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. S. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020, doi: [10.1109/TVT.2020.2982508](https://doi.org/10.1109/TVT.2020.2982508).
- [28] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 2, pp. 511–526, Mar. 2012, doi: [10.1109/tsmca.2011.2159586](https://doi.org/10.1109/tsmca.2011.2159586).
- [29] Y. Luo and X. Peng, "Global path planning of four-wheel mobile robot based on improved PSO," *Comput. Simul.*, vol. 37, no. 7, pp. 373–379, Jul. 2018.
- [30] M. Gao, H. Tang, and P. Zhang, "Survey of path planning technologies for robots swarm," *J. Nat. Univ. Defense Technol.*, vol. 43, no. 1, pp. 127–138, Feb. 2021, doi: [10.11887/j.cn.202101017](https://doi.org/10.11887/j.cn.202101017).
- [31] B. Huang, J. Yang, S. Lu, H. Chen, and D. Xie, "Wave capture power prediction based on improved particle group optimization neural network algorithm," *Acta Energiae Solaris Sinica*, vol. 42, no. 2, pp. 302–308, Feb. 2021, doi: [10.19912/j.0254-0096.tynxb.2018-0910](https://doi.org/10.19912/j.0254-0096.tynxb.2018-0910).
- [32] K. Kumar and S. Pabboju, "An efficient adaptive genetic algorithm technique to improve the neural network performance with aid of adaptive GA operators," *Int. J. Netw. Virtual Organisations*, vol. 20, no. 2, pp. 127–142, 2019, doi: [10.1504/IJNVO.2019.097630](https://doi.org/10.1504/IJNVO.2019.097630).
- [33] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 1559–1576, Feb. 2021, doi: [10.1007/s12652-020-02228-z](https://doi.org/10.1007/s12652-020-02228-z).



**WANG ZHIHENG** received the bachelor's degree in engineering from Southwest Petroleum University, in 2019. He is currently pursuing the master's degree in computer science with the Xi'an University of Posts and Telecommunications. His current research interests include biologically inspired algorithm and deep learning.



**LIU JIANHUA** was born in September 1963. He received the degree from the Beijing Institute of Posts and Telecommunications, in July 1986. He worked with the 10th Research Institute of the Ministry of Posts and Telecommunications, Xi'an, in 1986. He transferred to the Institute of Communication Technology, Xi'an University of Posts and Telecommunications, in September 1994. He has been engaged in scientific research and teaching with the Department of Information and Control and the School of Computer Science. He was appointed as the Director of the Information Security Laboratory, in 2004, the Deputy Director of Information and Control Department, in 2006, and the Vice President of the Computer Science College, in 2009. He is a Senior Engineer.

• • •