The Phasmatodea Population Evolution Algorithm and Its Application in 5G Heterogeneous Network Downlink Power Allocation Problem

Jeng-Shyang Pan[1,2], Pei-Cheng Song[1], Chun-An Pan[3], Ajith Abraham[4]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] College of Information Science and Technology, Dalian Maritime University, China
[3] School of Architecture, Design and Planning, University of Sydney, Australia
[4] Machine Intelligence Research Labs (MIR Labs), Auburn WA, USA
jspan@cc.kuas.edu.tw, spacewe@outlook.com, chunanpan@gmail.com, Ajith.abraham@ieee.org

Abstract

This paper proposes a new population-based heuristic algorithm called Phasmatodea (stick insect) Population Evolution (PPE) algorithm. The implementation of the algorithm is inspired by the evolutionary characteristics of stick insect population. Stick insects will change their shape to adapt to the surrounding environment, and stick insect populations will also evolve as the environment changes. This paper implements the PPE algorithm based on the relevant characteristics of the stick insect and the population model. The PPE algorithm is compared with the classic and newer algorithms on 30 benchmark functions. The results show that the proposed algorithm has strong competitiveness. Then this paper tries to apply the proposed algorithm to the downlink power allocation problem in the fifth-Generation (5G) heterogeneous network, and has achieved certain results.

Keywords: Phasmatodea Population Evolution algorithm, Two-layer heterogeneous network, Downlink power allocation problem, Population competition model, Population growth model

1 Introduction

Meta-heuristic algorithm is an effective and widely applicable stochastic optimization algorithm. It is widely used in industrial and economic issues to handle complex optimization problems [1-3]. Real-world engineering optimization problems are usually complex, and sometimes it is difficult to establish effective models to calculate. Since the meta-heuristic algorithm is usually easy to implement, suitable for different types of problems, and can achieve better results in a limited time, it has good application capabilities [4-7].

The proposed different meta-heuristic algorithms have achieved good results in various fields [8-12]. According to the source of the algorithm's inspiration, they can be divided into swarm-inspired, bio-inspired, evolution-inspired, physical-inspired and so on [13-14]. Evolution-inspired algorithms come from imitating the natural evolution of species. Genetic Algorithms (GA) was proposed earlier and is also a more classic algorithm [15].

Swarm-inspired algorithms are from the different behaviors of living things in nature, and then design different algorithms by imitating different animal behaviors. Among them, Particle Swarm Optimization (PSO) [16-17] and Ant Colony Optimization (ACO) [18-20] were proposed earlier, and they are also classic algorithms. In addition, other different algorithms have been proposed one after another, and have achieved good performance in different fields, such as Artificial Bee Colony (ABC) algorithm [21-22], Bat Algorithm (BA) [23-24], Cuckoo Search (CS) algorithm [25-27], Grey Wolf Optimizer (GWO) [28-30], Pigeon-Inspired Optimization (PIO) algorithm [31], etc. Physical-inspired algorithms are usually inspired by the phenomena and laws of the physical world. Algorithms are usually derived from existing physical or mathematical knowledge, classic and widely used algorithms include Simulated Annealing (SA) algorithm [32], Black Hole (BH) algorithm [33-34], Gravitational Search Algorithm (GSA) [35-36], Multi-Verse Optimizer (MVO) [37-38], Charged System Search (CSS) algorithm [39-40], States of Matter Search (SMS) algorithm [41], etc.

Although the above algorithms are different from each other, there are some common features among most of the algorithms. For example, the interaction between solutions usually occurs between the global optimal solution and the current solution, or the historical optimal interaction of the current solution, or the interaction with a better solution. The movement of an individual is usually affected by one or several good

*Corresponding Author: Pei-Cheng Song; E-mail: spacewe@outlook.com
DOI: 10.53106/160792642021112206001

solutions, in addition to which there is no basis and ability for autonomous decision-making. When individuals are not affected by other individuals, their movements are usually random, and existing information cannot be used effectively. In addition, there are fewer ways to interact between any two individuals.

In order to expand the research scope of related algorithms, this paper refers to the evolution characteristics of the stick insect population and proposes a Phasmatodea Population Evolution (PPE) algorithm to solve the optimization problem in the n-dimensional decision space. In the following statement, stick insect has the same meaning as Phasmatodea.

With reference to the characteristics of convergent evolution in the evolution of the stick insect population, the current stick insect population (i.e., a particle) tends to be the closest geographically optimal population rather than the global optimal population. The movement of the stick insect population is not only affected by the most nearest optimal population, but also changes its movement trend based on its previous movement trend and its own attributes. If the stick insect population obtains a better result in the evolution process of the previous step, it is more inclined to maintain the previous evolution path in the next evolution process. If the population of the current population is too small, it will be more conservative in the evolution process. On the contrary, the stick insect population will have a greater probability of variability and explore more areas. Based on the above methods, individuals have certain autonomous decision-making capabilities and can use existing information more effectively. In addition, the interaction between any two stick insect populations also has the characteristics of population competition, which increases the interaction between any two populations. More detailed information will be explained later. The performance of the PPE algorithm has been tested with various benchmark functions. The results show that the proposed algorithm has strong competitiveness and can also be effectively applied to some engineering optimization problems.

The article is organized as follows. Section 2 introduces the relevant knowledge required to develop the proposed PPE algorithm. Section 3 introduces the principle and implementation of the proposed PPE algorithm. Section 4 tests the performance of the algorithm and shows the results. Section 5 attempts to use the proposed PPE algorithm to deal with the downlink power allocation problem of fifth-Generation (5G) heterogeneous networks. Section 6 summarizes and discusses this paper.

2 Inspiration

2.1 Mimicry and Evolution

In the long process of species evolution, many species have developed different survival strategies. Stick insects have many interesting features and mimicry is one of the remarkable features [42-44]. Mimicry is the phenomenon of confounding the other party's cognition by mimicking the morphology or color of another species or environment during biological evolution, to protect itself or prey on other creatures. Stick insects in nature are shown in Figure 1.

Figure 1. Stick insect in nature (Photo by Chun-An Pan)

Like other species, different environments have different effects on the evolution of stick insects. Generally, stick insect populations in different geographical environments will also evolve into different forms [45-46]. The stick insect population will migrate to find the most suitable environment for survival. During the migration, the shape of the stick insect will change, and the characteristics of the stick insect will also change. In order to show the feature evolution process more intuitively, this paper maps n features of stick insects in n-dimensional space to a two-dimensional coordinate system.

As shown in Figure 2, the horizontal axis represents n features, which is equivalent to n dimensions. The vertical axis is abstracted as the quantized values of different features, which is equivalent to the value range of each dimension. The blue solid line represents the current feature state of the stick insect population, which is equivalent to the value of x in the current dimension. The red dotted line represents the feature state adapted to a better environment, which is equivalent to the optimal solution in the n-dimensional decision space. Then the stick insect population needs to evolve on different characteristics in order to achieve the best living conditions. For example, in feature 7 and feature 3, stick insect populations need to evolve in different directions, and slowly evolve to the best feature state.

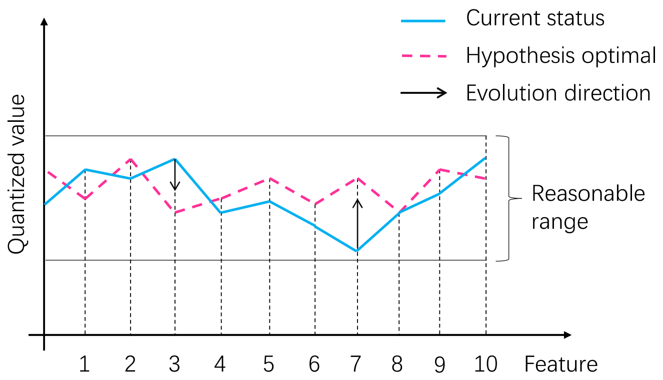


Figure 2. Evolution of a stick insect population

In this process, the stick insect population will seek the best living conditions. Similar evolution will also occur between different populations, but due to the different geographical distances of different populations, the stick insect population is more likely to have similar characteristics to the nearest dominant population, rather than other excellent populations far away. During the evolution of species, sometimes mutations that occur in individuals are preserved, and sometimes not, beneficial mutations are more likely to manifest through continuous accumulation.

2.2 Population Reproduction and Competition Model

The stick insect population also has two attributes: population quantity and population growth rate. During evolution, competition among populations will also occur. Different populations are also affected by changes in the environment. It is possible that the population will grow, or the population will decrease until it disappears.

When a population is not restricted by environmental factors, its population increases exponentially. However, in the real environment, the growth of the population is limited by resources and space. The population growth model is usually given using the logistic equation.

$$\frac{dp}{dt} = rp\left(1 - \frac{p}{k}\right) \quad \textbf{(1)}$$

In Equation (1), p represents the population quantity, r represents the effective growth rate of the population. K is the maximum environmental capacity of the species. As p increases, r will continue to decrease, and eventually when p reaches K, the population will stop growing.

In the PPE algorithm, this paper uses the logistic difference equation, let K=1 [47]. Then using the population quantity at time t can calculate the population quantity at time t+1 as

$$p^{t+1} = ap^t(1 - p^t) \quad \textbf{(2)}$$

In Equation (2), the value of p is between 0 and 1, a is the growth rate, and the value of a is between 0 and 4. When a exceeds the interval 0 to 4, it is meaningless. If $a < 1$, it means that the population quantity p is gradually decreasing, and eventually the population quantity will decrease to 0, and the population will disappear. If $1 < a < 3$, the population quantity will slowly converge to a stable value $p = (a-1)/a$. If $a > 3$, the population quantity will become unstable.

When competition occurs between two populations, the growth formula of the population becomes different.

$$\frac{dp}{dt} = r_1 p\left(1 - \frac{p}{n_1} - s_1\frac{q}{n_2}\right) \quad \textbf{(3)}$$

In Equation (3), q represents the population quantity of another population. r_1 represents the effective growth rate of the population p. n_1 is the maximum environmental capacity of population p, n_2 is the maximum environmental capacity of population q. s_1 means that relative to the resources supporting p, a unit quantity of q consumes s_1 times more resources than a unit quantity of p. The symbols used in this section are only used to introduce the knowledge of the evolution of related biological populations, and have nothing to do with the symbols in the algorithm formulas mentioned later.

3 Phasmatodea Population Evolution Optimization Algorithm

In order to deal with the optimization problem in n-dimensional space, this paper introduces the process of PPE algorithm. The algorithm mainly includes initialization, population position update and population evolution trend update. Among them, population evolution trend is affected by population evolution results and population competition.

3.1 Initialization

Suppose $\mathrm{x} = [x_1, ..., x_i, ..., x_n]$ is used to denote a position in n-dimensional space, and it also means a stick insect population, and its position also represents n features of the population.

Each solution x represents a stick insect population, and each x has two attributes, namely the population quantity p and population growth rate a. First generate Np solutions randomly, using the following formula.

$$x_i = Lb + (Ub - Lb) \cdot rand \quad \textbf{(4)}$$

Lb and Ub are the lower limit and upper limit of n-dimensional space **Ω**, $Lb = [lb_1, ..., lb_n]$, $Ub = [ub_1, ..., ub_n]$. $rand$ is an n-dimensional random vector, and the value of each dimension ranges from 0 to 1. x_i represents the i-th solution, $i \leq Np$. The population quantity of each solution x_i is initialized to

$$p_i = \frac{1}{Np} \quad \textbf{(5)}$$

Equation (5) is used to initialize the population quantity of each stick insect population x to ensure that the initial states of different stick insect populations are the same. Because the initial positions of different stick insect populations may be different, it is difficult to set an appropriate population quantity based on the initial position. The population quantity p needs to be updated according to the evolution of the population.

The population growth rate is initialized to $a = 1.1$. *ev* represents the current evolution trend of stick insect population, initialized to ev=0. This ensures that each randomly generated population can survive and reproduce normally at the beginning. Then calculate the fitness value of each solution, find the solution with the best fitness value as the global optimal solution *gbest*. Geographic location and environment will have a great influence on the evolution of stick insects. For a stick insect population, its morphology and habits are more similar to other populations with closer geographical positions.

So this article selects k historical optimal solutions to guide the movement of the surrounding solutions, stored in *Ho*, $Ho = [x_{h1}, ..., x_{hi}, ..., x_{hk}]$. Among them, x_{hi} is the i-th optimal solution that has appeared, x_{h1} is the optimal solution of all the solutions that have appeared, and x_{h2} is the optimal solution that has appeared only after x_{h1}. If a total of Np solutions are generated, then the number of k is

$$k = [\log(Np)] + 1 \quad \textbf{(6)}$$

Ho uses the Np solutions generated by the initialization process to select k optimal solutions, and its fitness value is stored in the corresponding array.

3.2 Population Position Update

After the initialization is completed, the algorithm begins to enter the iterative process, which is equivalent to the stick insect population starting to find a better habitat, and it will also change its position and shape to adapt to the environment. The new position is obtained based on the current position plus the evolution trend, the formula is

$$x^{t+1} = x^t + ev \quad \textbf{(7)}$$

In Equation (7), *ev* represents the current evolutionary trend of stick insect populations. t is the current number of iterations. After the population position changes, it is necessary to re-evaluate all solutions, update the global optimal solution *gbest* and k optimal solutions *Ho*.

3.3 Population Evolution Trend

Population evolution trends include the evolution direction formed by itself and the impact of population competition, and natural selection may also lead to population disappearance.

During the continuous migration of stick insect populations, the environment is also changing, and its own shape must constantly adapt to the environment. During the migration of stick insect populations, it may migrate to a better environment, and may migrate to a worse environment. Changes in the environment will affect the number of populations and the next evolutionary trend.

Corresponding to Np solutions in n-dimensional space, the moving result of each solution is different. This article discusses two cases, and different evolutionary operations will be taken in different cases.

The first case is that the fitness value becomes better after the solution moves, then this change will affect the current solution population and growth rate. The effect on the growth rate can be written as

$$a^{t+1} = a^t(1 + \frac{f(x^t) - f(x^{t+1})}{f(x^{t+1})}) \quad \textbf{(8)}$$

For a certain solution x^t, x^t represents a stick insect population when the number of iterations is t. The position of the solution x^t is updated to x^{t+1}, and the calculated fitness values of the two are $f(x^t)$ and $f(x^{t+1})$. Equation (8) is used to update the growth rate of the population. If $f(x^{t+1})$ is better than $f(x^t)$, then a^t will increase, indicating that the direction of evolution is beneficial to population survival. Otherwise, a^t will decrease, indicating that the effect after the location update is not good, and the population growth rate will also decrease. Then, the effect on the population can be written as

$$p^{t+1} = a^{t+1} p^t (1 - p^t) \quad \textbf{(9)}$$

After updating the population number and growth rate, it is necessary to update the evolution trend *ev*. The evolution trend is affected by three aspects. The first aspect is similar evolution. The movement of the current solution will tend to the nearest optimal solution. Considering that different populations are constrained by geographic distance and sometimes do not know the information of the optimal population, this article uses the closest one in the *Ho* to the solution x to guide the movement of x. This method is conducive to exploring different local optimal regions. The second aspect is the continuation of the evolutionary trend. If the fitness value obtained after the movement is better, then continue this movement trend. For stick insect populations, this means that the evolution direction is correct, and the populations can obtain a better living environment.

The third aspect is mutation within the population. Mutation adds uncertainty to evolution, and at the same time, small mutations can grow to an extent that affects evolutionary trends after a long period of accumulation. Based on the above three aspects, the updated formula of population evolution trend is

$$ev^{t+1} = (1-p^{t+1})\left(s(Ho,x^t)-x^t\right)\cdot\alpha + p^{t+1}ev^t + \beta\cdot p^{t+1}m \quad \textbf{(10)}$$

In Equation (10), $s(Ho, x^t)$ is used to find the historical optimal solution that is closest to x^t in Ho. α and β are adjustment parameters, used to adjust the influence weight of different items. m is used to represent small mutations in the population, in this paper m=0.1 $\cdot$ rdn $\cdot$ (Ub-Lb). rdn is an n-dimensional vector, each dimension is randomly generated using a standard normal distribution. Equation (10) is used to deal with the situation where a better fitness value is obtained after moving, but sometimes it is not possible to obtain a better fitness value after moving.

The second case is that the fitness value becomes worse after moving, then there is a certain probability that the solution after moving will be accepted.

If the stick insect population accepts the current poor position, it will affect the population quantity and growth rate, and the update methods of the two are the same as Equation (8) and (9). The acceptance probability used in this article is p, that is, the quantity of populations. First, a uniform distribution is used to generate a random number from 0 to 1. If it is less than p, then the poor position is accepted, otherwise it is not accepted.

Next, the evolution trend needs to be updated. Because the result becomes worse after the move, then the previous evolution trend will not be continued, but the exploration will be conducted around the current solution to find a better evolution direction. The update formula is

$$ev^{t+1} = \left(s(Ho,x^t)-x^t\right)\cdot rand + st\cdot rdn \quad \textbf{(11)}$$

In Equation (11), st is used to control the exploration range of the surrounding area, and its calculation formula is

$$st^{t+1} = \left(\frac{1}{t}\right)^{\frac{|f(x^{t+1}-f_{min}^{t+1})|}{f_{min}^{t+1}-f_{max}^{t+1}}}\cdot c^{t+1} \quad \textbf{(12)}$$

In Equation (12), c is set to 2 during the initialization process, and c gradually decreases after entering the iterative process, and the updated formula is c^{t+1}=0.99$\cdot c^t$. Based on the above description, the influencing factors of population evolution trend can be represented by Figure 3.

As shown in Figure 3, x^t obtains a better fitness value after moving, so the next evolution trend ev^{t+1} continues the movement trend of ev^t, and adds the influence of the closest optimal solution x_{hk} and mutation m. x_j does not obtain a better fitness value after moving, so it is necessary to randomly explore around to find a better direction of evolution.

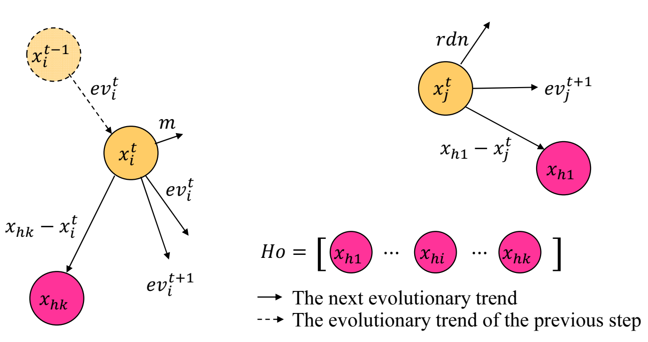


Figure 3. Different influencing factors of population evolution trend

In addition to the evolution of the population itself, competition between the populations will also affect the evolutionary trend of the population, so after completing the above steps, the algorithm needs to add the influence of competition between the populations. The population competition model is shown in Equation (3). Equation (3) is modified in this paper and used to update the population quantity.

$$p_i = p_i + a_i p_i (1 - p_i - \frac{f(x_j)}{f(x_i)} p_j) \quad \textbf{(13)}$$

When the population x_i is competing with x_j, the population quantity of x_i is calculated using Equation (13). Compared to Equation (3), both n_1 and n_2 are 1, and s_1 is calculated as the ratio of the fitness values between the x_i and x_j.

The population growth rate is not updated at this time. After updating the population quantity, the evolutionary trend of the solution needs to be updated

$$ev^{t+1} = ev^{t+1} + \frac{f(x_j)-f(x_i)}{f(x_j)}(x_j - x_i) \quad \textbf{(14)}$$

As shown in Equation (14), if the fitness value $f(x_i)$ of the population x_i is less than $f(x_j)$, it means that the population x_i has stronger competitiveness, and the population x_i will move toward x_j and occupy the space of x_j, otherwise it will move in the opposite direction to stay away from x_j.

Before calculating the results of population competition, it is necessary to determine whether the two populations have competed.

For the solution x_i, this paper randomly selects a solution x_j from the other Np-1 solutions. If the distance between the two solutions is less than G, it means that the survival spaces of the two populations overlap and competition will occur. The calculation method of G is G =0.1(ub-lb)(($Maxgen$+1-t)/$Maxgen$). $Maxgen$ is the maximum number of iterations. Then use Equations (13) and (14) to calculate the result of the competition.

In addition, when the population is too small or the population growth rate exceeds the normal range, the solution needs to be eliminated, and a new solution is generated to replace, indicating that the current stick insect population disappears due to poor adaptability to the environment.

Based on the above description, the pseudo-code of the PPE algorithm is shown in Algorithm 1.

Algorithm 1. PPE

1. Initialize *Np* solutions using Eq (4);
2. Initialize *ev, p, k* and *a* using Eq (5), (6);
3. Calculate fitness $f(x)$, set *gbest* and *Ho*;
4. for $t = 2$ *to Maxgen* do
5.   Update each x to *newx* using Eq (7);
6.   Calculate new fitness $f(newx)$,
  update *gbest* and *Ho*;
7.   for $i = 1$ *to Np* do
8.   Update a_i and p_i using Eq (8) and (9);
9.   if $f(newx) \le f(x)$ then
10.   update x, $x = newx$, update $f(x)$;
11.   Update ev_i using Eq (10);
12.   if $f(newx) > f(x)$ then
13.   if $rd < p_i$ then
14.   Update x, $x = newx$, update $f(x)$;
15.   Update ev_i using Eq (11) and (12);
16.   Randomly choose a solution x_j, $(j \ne i)$;
17.   if $dist(x_j, x_i) < G$ then
18.   Update p_i using Eq (13), update ev_i using
      Eq (14);
19.   if $p_i \le 0$ *or* $a_i \le 0$ *or* $a_i > 4$ then
20.   Eliminate x_i and replace it using Eq (4);

In Algorithm 1, *rd* is a random number generated using a uniform distribution, and the range is 0 to 1. The *dist*() function is used to calculate the Euclidean distance between two *n*-dimensional vectors. Algorithm code is exposed on the website. (https://github.com/Spacewe-outlook/PPE)

Based on the above description, the flowchart of Algorithm 1 is shown in Figure 4.

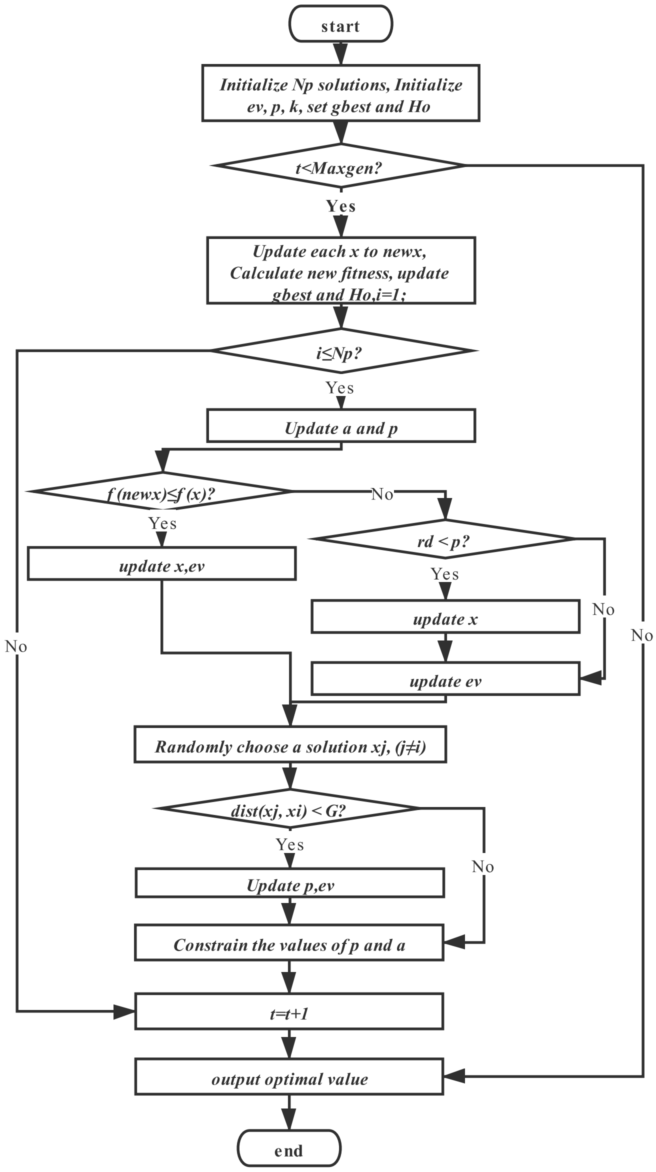


Figure 4. Flow chart of PPE algorithm

4 Experimental Results

4.1 Parameter Arrangement

In order to compare the overall performance of the PPE algorithm, this paper tests the proposed algorithms. There are many test functions for related algorithms, but there are some common test suites in the CEC competition. This paper uses CEC2014 [48]benchmark suites for testing. All the benchmark functions include 3 unimodal functions (F1-F3), 13 simple multimodal functions (F4-F16), 6 mixed functions (F17-F22) and 8 composition functions (F23-F30). This paper uses a variety of algorithms for comparison, including classic PSO [16], GA [15], SA [31], GSA [34], GWO [28], BOA [49] and ASO [50] algorithms. The parameter settings of related algorithms are shown in Table 1.

Table 1. Parameter settings

Algorithm	Parameter
PSO	Np = 20, c1 = 2, c2 = 2, Inertia constant: 0.8 to 0.2 linearly decreasing;
GA	Np = 20, Mutation Rate = 0.1, Crossover Rate = 0.4;
SA	Np = 1, Initial Temp = 0.1, Temp Reduction Rate = 0.99, Mutation rate = 0.5;
GSA	Np = 20, Initial gravitational constant = 100, Decreasing coefficient = 20;
GWO	Np = 20;
BOA	Np = 20, Probability switch = 0.8, Modular modality = 0.01, Power exponent = 0.1;
ASO	Np = 20, Depth weight = 50, Multiplierweight = 0.2;
PPE	Np = 20, $\alpha = 0.2$, $\beta = 2$;

Table 1 shows the parameter settings of different algorithms and the number of solutions *Np* generated in each iteration.

4.2 Comparison with Other Algorithms

For a fairer comparison, the function evaluation times of all algorithms need to be the same, so the total function evaluation times of each algorithm is 40000. All benchmark functions have a dimension of 30. Test each algorithm 11 times in each test function, and record the average, standard deviation, and best values. At the same time, the overall performance of each algorithm compared with the PPE algorithm was measured at a significant level α = 0.05 under the Wilcoxon's sign rank test.

Table 2 shows the comparison test results of PPE algorithm with other algorithms. The test results of each algorithm on each function are compared with the results of the PPE algorithm. The symbol (<) indicates that the algorithm does not perform as well as the PPE algorithm on the current function. The symbol (>) indicates that the PPE algorithm performs poorly. The symbol (=) indicates that the two algorithms perform similarly on the current benchmark function. The comparison results on all benchmark functions are summarized in the last row of the table. According to the data in Table 2, the proposed PPE algorithm can obtain better results than other algorithms. Compared with the PSO algorithm, the PPE algorithm achieves better results on 16 benchmark functions. Compared with the newer ASO algorithm, the PPE algorithm only achieves better results on 15 benchmark functions, and the results on the remaining 15 benchmark functions are not good. Compared with the newer BOA algorithm, the PPE algorithm achieves better results on 23 benchmark functions, but the results on the composition function are not good. And the results of the BOA algorithm found on the functions F29, F30 are far better than other algorithms. There are 30 benchmark functions used in this paper, which are divided into four types. Based on the data in Table 2, this paper compares the performance of all algorithms on different types of functions. The results are shown in Table 3.

Table 2. The results of experiments

Function type	PSO	GA	ASO	BOA	GSA	GWO	SA	PPE
F1	2.58E+0.7(<)	2.96E+0.7(<)	1.62E+0.7(<)	1.37E+0.9(<)	1.81E+0.7(<)	9.2E+0.7(<)	4.81E+0.7(<)	8.36E+0.6
F2	8854.18(>)	4.87+0.7(<)	9337.37(>)	7.13E+10(<)	7554.07(>)	4.01E+0.9(<)	2.13E+0.9(<)	2.87E+0.5
F3	5033.52(<)	14522.59(<)	6377.48(<)	82152.84(<)	54976.95(<)	42123.96(<)	10465.25(<)	2083.88
F4	600.43(<)	562.20(<)	628.11(<)	16447.25(<)	866.58(<)	747.19(<)	670.12(<)	529.20
F5	520.99(<)	520.48(<)	520.00(>)	521.07(<)	520.00(>)	521.02(<)	520.94(<)	520.06
F6	613.90(>)	619.35(>)	610.05(>)	638.90(<)	621.73(<)	616.62(>)	625.10(<)	620.54
F7	700.01(>)	701.51(<)	700.00(>)	1554.64(<)	700.00(>)	724.66(<)	719.73(<)	700.25
F8	840.88(>)	820.82(>)	867.30(>)	1117.64(<)	948.97(<)	904.83(<)	940.41(<)	900.24
F9	985.72(>)	1031.98(>)	973.90(>)	1254.21(<)	1068.96(<)	1017.16(>)	1136.82(<)	1042.85
F10	2176.83(>)	1976.56(>)	3677.30(<)	8819.12(<)	4055.22(<)	37.32.28(<)	5086.89(<)	2985.00
F11	4489.98(>)	4644.42(<)	4454.58(>)	9066.16(<)	5334.09(<)	4130.50(>)	7638.82(<)	4607.05
F12	1202.12(<)	1200.55(<)	1200.05(>)	1203.47(<)	1200.01(>)	1201.96(<)	1202.25(<)	1200.34
F13	1300.59(<)	1300.51(<)	1300.42(<)	1309.34(<)	1300.38(<)	1300.67(<)	1300.83(<)	1300.38
F14	1400.29(<)	1400.39(<)	1400.34(<)	1698.83(<)	1400.28(<)	1405.68(<)	1401.59(<)	1400.24
F15	1512.17(>)	1531.57(<)	1503.35(>)	4.69E+0.5(<)	1506.84(>)	1712.81(<)	1551.85(<)	1524.07
F16	1612.13(>)	1611.93(>)	1612.65(<)	1613.30(<)	1613.40(<)	1611.61(>)	1612.83(<)	1612.63
F17	1.12E+0.6(<)	2.74E+0.6(<)	1.12E+0.6(<)	2.27E+0.8(<)	7.54E+0.5(<)	5.57E+0.6(<)	2.29E+0.6(<)	5.70E+0.5
F18	6984.93(<)	10290.43(<)	35.19.88(<)	.521E+0.9(<)	2863.81(>)	1.75E+0.7(<)	6.67E+0.5(<)	3216.12
F19	1909.03(>)	1947.30(<)	1950.90(<)	2420.66(<)	2013.80(<)	1964.33(<)	1917.60(>)	1936.14
F20	7936.99(<)	44979.43(<)	24458.87(<)	5.30E+0.6(<)	69617.54(<)	36872.34(<)	6629.40(>)	6971.38
F21	4.65E+0.5(<)	1.09E+0.6(<)	3.07E+0.5(<)	3.89E+0.7(<)	2.74E+0.5(<)	8.36E+0.5	7.50E+0.5(<)	1.65E+0.5
F22	2555.52(>)	2887.69(<)	2694.09(>)	83695.37(<)	3306.90(<)	2637.11(>)	2792.23(>)	2828.51
F23	2616.36(<)	2615.72(<)	2620.58(<)	2500.00(>)	2620.38(<)	2647.34(<)	2624.53(<)	2615.62
F24	2639.93(<)	2632.44(<)	2625.43(>)	2600.00(>)	2620.18(>)	2600.02(>)	2658.27(<)	2628.03
F25	2712.23(>)	2715.23(<)	2710.99(>)	2700.00(>)	2707.29(>)	2709.93(>)	2712.84(>)	2714.15
F26	2740.59(>)	2755.10(>)	2701.52(>)	2767.42(>)	2800.07(<)	2745.88(>)	2728.90(>)	2782.00
F27	3342.75(<)	3618.08(<)	3144.57(>)	3669.88(<)	4494.99(<)	33.91.49(<)	3483.30(<)	3323.76
F28	3992.89(>)	4927.99(>)	6251.65(>)	5610.50(>)	6893.21(<)	4189.27(>)	4340.26(>)	6286.50
F29	2.24E+0.6(<)	4943.81(>)	3.27E+0.7(<)	3100.00(>)	31454.50(>)	1.69E+0.6(<)	8.61E+0.5(<)	7.28E+0.5
F30	12609.93(<)	14048.25(<)	19355.06(<)	3200.00(>)	1.08E+0.8(<)	93492.58(<)	11114.87(<)	10073.14
</=/>	16/0/14	22/0/8	15/0/15	23/0/7	21/0/9	21/0/9	24/0/6	

Table 3. Performance of different algorithms on different types of functions

Function type	PSO	GA	SA	GSA	GWO	BOA	ASO	PPE	
Unimodal	0	0	0	1	0	0	0	2	3
Multimodal	0	2	0	2	2	0	4	3	13
Hybrid	2	0	1	1	0	0	0	2	6
Composition	1	0	0	0	0	5	2	0	8
Win	3	2	1	4	2	5	6	7	30

Table 3 compares how many functions the current algorithm has achieved the best results in each type of function compared to all other algorithms.

In Table 3, there are three unimodal functions. The result of the GSA algorithm in one of the functions is better than all other algorithms, so the GSA algorithm has a value of 1 in this type of benchmark function. The result of the proposed PPE algorithm in two unimodal functions is better than all other algorithms, so the value is 2. The last row of Table 3 counts the number of algorithms that have achieved the best results on all functions. The PPE algorithm achieved the best results among the 7 functions, followed by the ASO algorithm, which achieved the best results among the 6 functions. It is worth noting that the proposed PPE algorithm does not have better results than other algorithms in the eight composition functions. Although the BOA algorithm does not work well in other types of functions, it has achieved the best results among the 5 composition functions.

4.3 Qualitative Analysis

In order to more intuitively show the search mode of the proposed algorithm on different types of functions, this article randomly selected 6 benchmark functions in CEC2014, as shown in Figure 5. The functions selected in this paper are F1, F6, F8, F13, F26, F27. All functions have a dimension of 2. The number of iterations of the algorithm is 100, and the *Np* is 20.

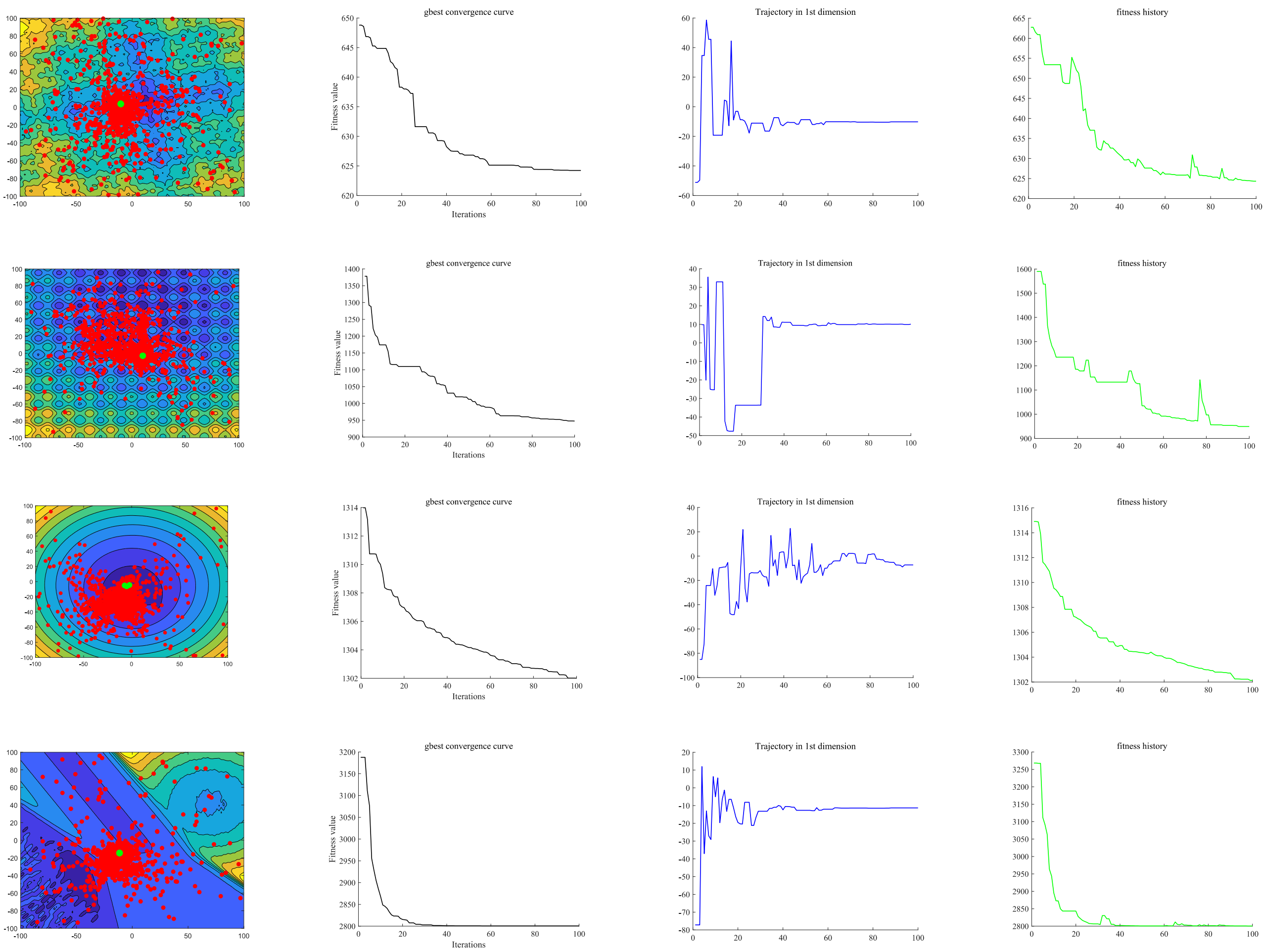


Figure 5. Qualitative results in different functions

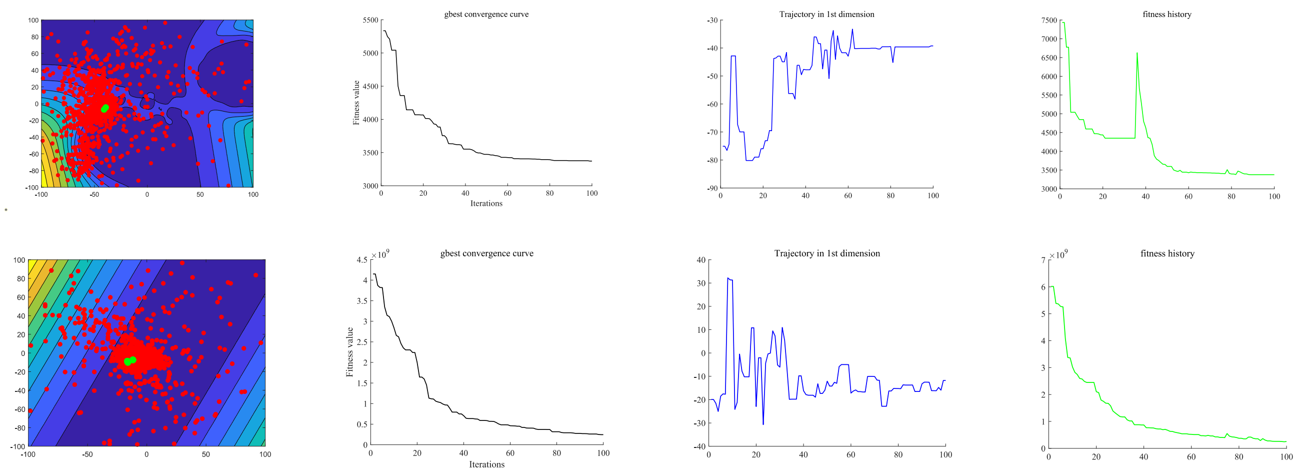


Figure 5. Qualitative results in different functions (continue)

The first column shows the distribution of all searched positions during the iteration. The red dot indicates the position of a solution, and the green dot indicates the final position of k solutions in *Ho*. The second column shows the convergence process of the fitness value of the optimal solution *gbest*. The third column shows the changing process of the value of the first dimension of the optimal solution *gbest*. In Figure 5, the first dimension of the optimal solution fluctuates greatly at the beginning, and then slowly converges to a stable interval. The fourth column shows the change process of the fitness value of the first solution. Combined with the characteristics of the algorithm, each solution generally tends to find a better position, so the fitness value in the fifth column is generally smaller. However, for a particular solution, this solution may accept a poor position, so there may be a sudden increase in the fitness value during the decline of the fitness value. But the general trend of solutions tends to be smaller fitness values.

In addition, in order to show the changes of the parameters during the operation of the algorithm more clearly, this article provides a display of the changes in the population quantity p of the algorithm on benchmark function F6, as shown in Figure 6.

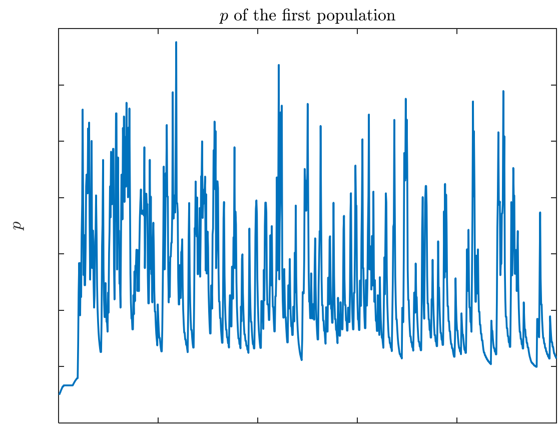


Figure 6. Population size p of the first solution x on F6

4.4 Convergence Evaluation

For meta-heuristic algorithms, the algorithm needs multiple iterations to find the optimal value. Different algorithms may find similar optimal solutions after the same number of iterations or evaluations, which means that different algorithms may eventually obtain similar solutions. However, the search process of different algorithms is different, and the speed and method of converging to the global optimum are also different. Therefore, it is necessary to evaluate the convergence of the algorithm.

The convergence curves of the proposed PPE algorithm and other algorithms on some benchmark functions are shown in Figure 7. This article selected several different types of benchmark functions for display. As shown in Figure 7, the horizontal axis represents the number of function evaluations. In order to show the difference between different curves more clearly, the vertical axis is *Log(f - f*)*. *f* represents the final result obtained by the algorithm on a benchmark function. *f** represents the minimum value of the current benchmark function.

As shown in Figure 7, the proposed algorithm achieves better results on the functions F1, F13, F14 and F21, and its convergence ability and final result are better than other algorithms. The result on F27 is worse than the ASO algorithm, but slightly better than other algorithms. The result on function 30 is worse than the BOA algorithm, and the convergence speed is relatively slow, but the final result is slightly better than other algorithms. Although the algorithm does not achieve the best results on the F5 and F10 functions, it has continuous convergence. Algorithms such as SA and BOA are more likely to fall into the local optimal on the F5 and F10 functions.

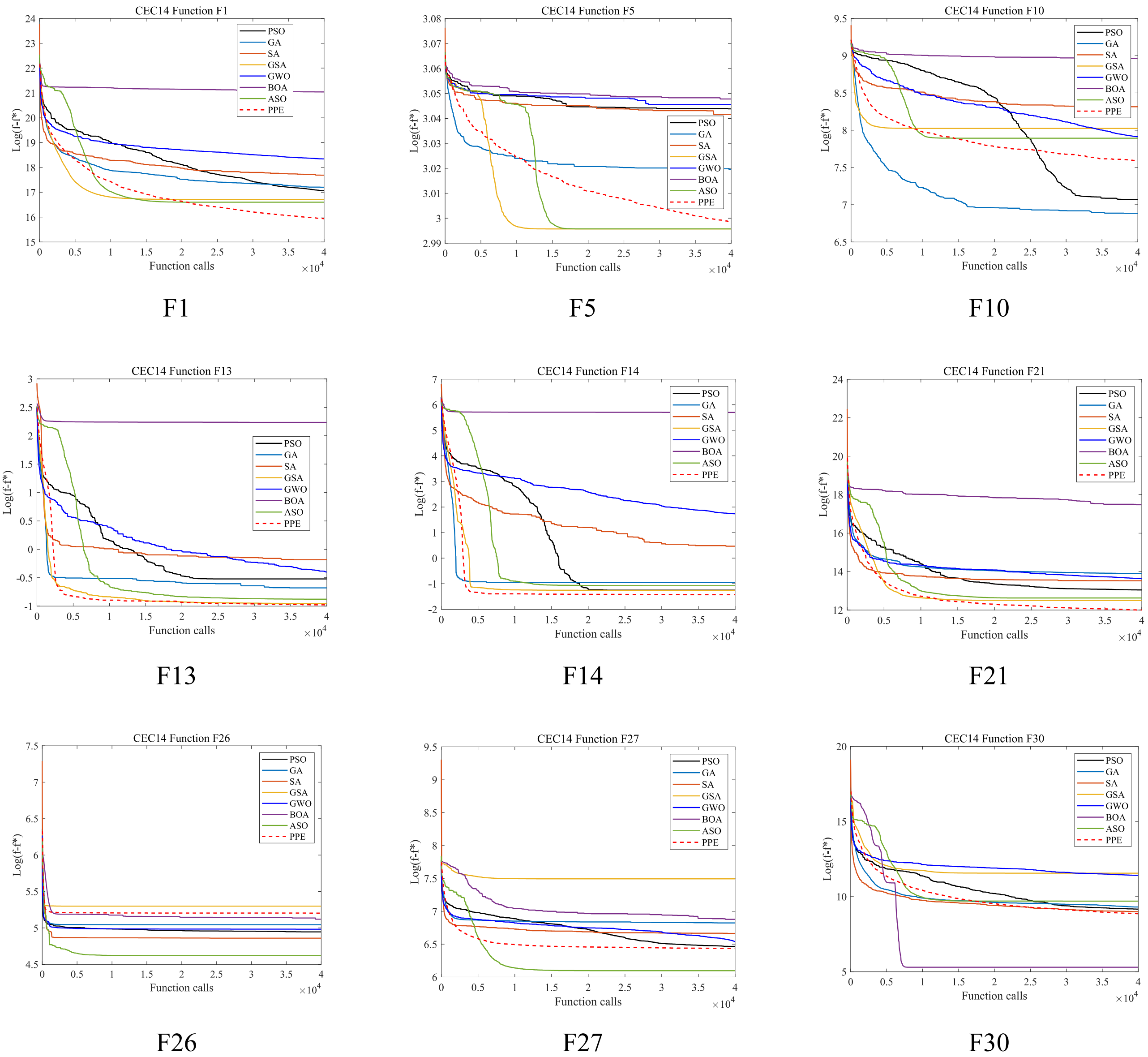


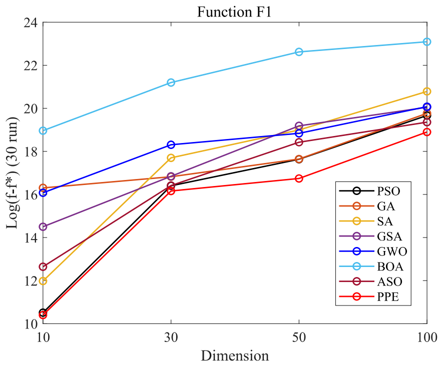
Figure 7. Convergence test results in 30 dimensions

4.5 Scalability Analysis

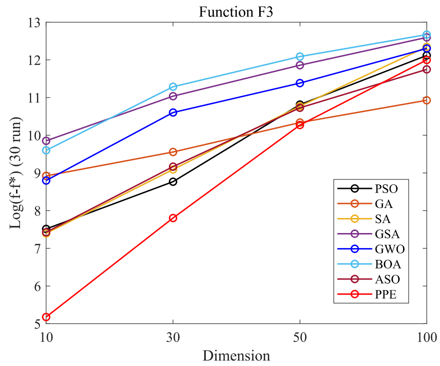
In order to test the performance of the proposed algorithm in different dimensions, this paper uses 30 benchmark functions to test the performance of the proposed algorithm and related algorithms at 10, 30, 50 and 100 dimensions.

In each dimension, each algorithm runs 30 times on each benchmark function, and the average value is recorded. The results of each algorithm in different dimensions on different benchmark functions are shown in Figure 8.

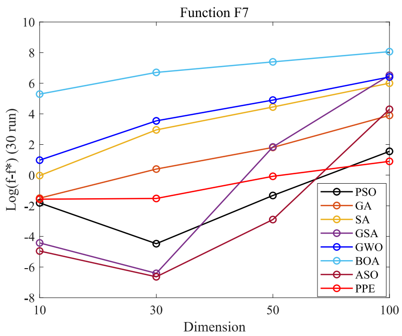
Figure 8 shows the overall performance of the proposed PPE algorithm on different benchmarks with different dimensions. As the dimension increases, the search becomes more complicated, and the fitness value that the algorithm can obtain is also increasing. The PPE algorithm proposed in the F1 and F17 has achieved the best results in different dimensions. For the function F18, although the proposed PPE algorithm is inferior to other algorithms in 10 and 30 dimensions, it can obtain the best results in 50 and 100 dimensions. For F14, the result of the proposed PPE algorithm when the dimension changes does not change much, and it can handle low-dimensional and high-dimensional situations better. For F7, compared with other algorithms, the performance of algorithm PPE in different dimensions is at a medium level. Combined with the data in Table 3, the performance of the PPE algorithm on the composition function F29 is not as good as that of the BOA algorithm.



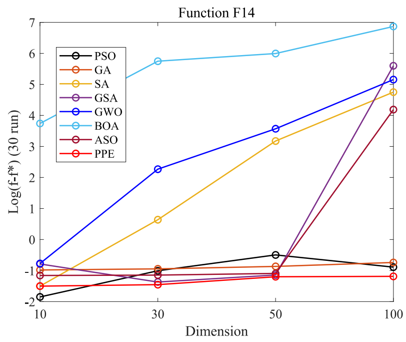
F1



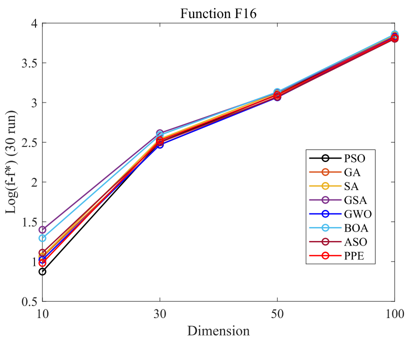
F3



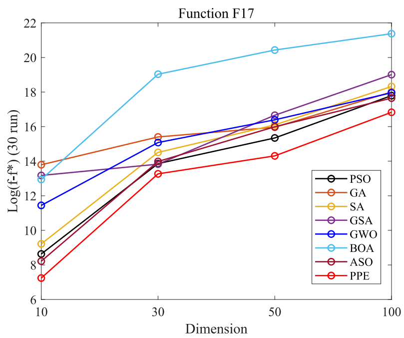
F7



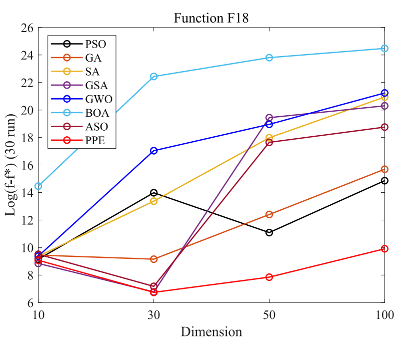
F14



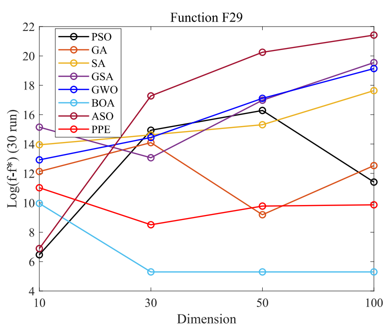
F16



F17



F18



F29

Figure 8. Comparison of different algorithms with different dimensions in different functions

5 PPE for Downlink Power Allocation Problem

In this section, a 5G heterogeneous network model is constructed to test the proposed algorithm. The model is a two-layer network of macrocell and femtocell. Macrocell Base Station (MBS) is used to achieve large-scale coverage, and Femtocell Base Station (FBS) is used to serve dense users and targeted coverage [51]. As a new heterogeneous network, macrocell/femtocell can effectively increase coverage and increase throughput, but it requires reasonable allocation of resources to improve efficiency.

There have been many related studies on the optimization of 5G heterogeneous networks, including centralized and distributed methods, game theory, etc. [52-55]. This paper uses the PPE algorithm to implement the downlink channel and power allocation, which is a centralized method. In the test results, the network Energy Efficiency (EE) is used as the evaluation standard. The calculation method is shown in Equation (15).

$$EE = C / P \qquad \textbf{(15)}$$

Where C represents the throughput of the entire heterogeneous network, which is the sum of the throughput of each user served by the macrocell base station and the femtocell base station. P represents the total energy consumed by all base stations in the entire two-layer heterogeneous network, including the resulting loss.

$$C_i = \frac{B}{K}(1 + log_2(\frac{|g_i\sqrt{p_i l_i}|^2}{\sum_w |g_i^w \sqrt{p_i^w l_i^w}|^2 + P_0})) \qquad \textbf{(16)}$$

Equation (16) is used to calculate the throughput of the i-th user, where B is the bandwidth and K is the number of sub-channels. g_i is the channel gain between the base station and the i-th user currently serving. p_i is the power allocated by the base station to the i-th user currently served. l_i is the path loss between the base station and the i-th user currently served. g_i^w is the channel gain between the interfering base station and the i-th user. pi is the power allocated by the interfering base station to the interfering user. l_i^w is the path loss from the interfering base station to the i-th user. w means that all interference to the i-th user must be calculated. P_0 is additive white Gaussian noise.

The model proposed in this paper has some limitations, including MBS and FBS share all frequency bands, only the downlink interference is considered. Assuming that there is no interference between different users served by the same base station, there will be interference only when users served by other base stations use the same channel as the current user.

Based on the above description, this paper constructs the minimization problem. In the two-layer heterogeneous network constructed in this paper, the number of MBS is 1, serving M Macrocell Users (MUE), the number of femtocell base stations is N, and each femtocell serves N_f Femtocell Users (FUE). The objective function is as follows.

$$\begin{aligned}
&Consider \quad \vec{x} && = [\vec{c}_{mue}, \vec{c}_{fue}, \vec{p}_{mue}, \vec{p}_{fue}];\\
&\vec{c}_{mue} && = [cm_1, ..., cm_i, ..., cm_M],\\
&\vec{c}_{fue} && = [cf_1, ..., cf_i, ..., cf_{N*N_f}],\\
&\vec{p}_{mue} && = [pm_1, ..., pm_i, ..., pm_M],\\
&\vec{p}_{fue} && = [pf_1, ..., pf_i, ..., pf_{N*N_f}]\\
&\min \quad f(\vec{x}) && = \frac{1}{EE}\\
& && = \frac{\sum \vec{p}_{mue} + \sum \vec{p}_{fue}}{\sum_1^M C_{mue}^i + \sum_1^{N*N_f} C_{fue}^j}\\
&st. \quad y_1(\vec{x}) && = \sum \vec{p}_{mue} - P_{MBS} + P_{cl} \le 0,\\
&y_2(\vec{x}) && = \sum_1^{N_f} pf_i - P_{FBS} + P_{cl} \le 0,\\
&Variable\ range && 0.5 < cm_i < K + 0.5\\
& && 0.5 < cf_i < K + 0.5
\end{aligned}$$

Where P_{MBS} represents the total power of a macrocell base station, and P_{FBS} represents the total power of a femtocell base station. P_{cl} means circuit loss. cm_i means the channel assigned to the i-th MUE, cf_i means the channel assigned to the i-th FUE. pm_i represents the power allocated to the i-th MUE, pf_i represents the power allocated to the i-th FUE. The overall network structure is shown in Figure 9.

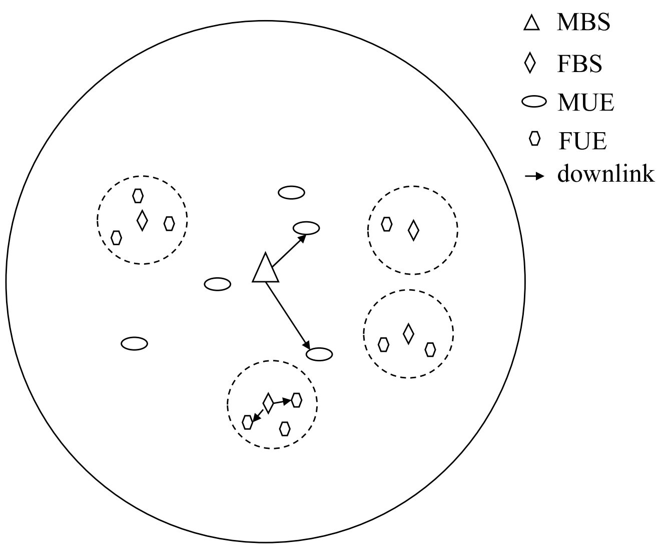


Figure 9. 5G heterogeneous network components

First, generate a heterogeneous network model, and then use the proposed algorithm to optimize the generated heterogeneous network. The relevant parameters are shown in the Table 4. Where MBS_{radius} and FBS_{radius} represent the coverage radius of MBS and FBS, respectively.

Table 4. Experimental settings for parameters

Symbol	Parameter
MBS_radius	600m
M	[10, 15, 20, 25, 30, 35]
P_MBS	600mW
FBS_radius	60m
N	[1, 2, 3, …, 19, 20]
Nf	[3, 4, 5]
P_FBS	500mW
B	100MHz
K	40
Pcl	0.5mW

This paper compares the optimization effect of the algorithm with different numbers of MUE and FUE. The results are shown in Table 5. A different heterogeneous network model is randomly generated each time, and each algorithm generates an initial solution randomly each time. The same model is tested five times, and the average value is taken and recorded. During the test, the function evaluation times of all algorithms are the same.

Table 5. Test results with different model parameters

M	N	Nf	GSA	BOA	ASO	PPE
10	2	5	348.38	322.4	301.17	285.98
10	5	5	75.768	48.909	67.891	64.117
10	10	3	115.07	101.94	104.37	101.32
10	10	4	153.4	128.44	136.34	131.09
10	10	5	76.546	57.115	70.801	68.706
15	2	5	195.23	171.97	179.44	167.00
15	5	5	216.19	201.99	195.72	192.56
20	2	5	263.39	259.36	237.26	235.60
20	5	5	259.15	228.95	237.92	226.77
25	2	5	172.11	155.90	158.50	151.86
25	5	5	201.04	180.01	185.08	182.15
30	2	5	101.20	77.62	92.37	89.17
30	5	5	71.577	51.753	65.089	64.782
35	2	5	163.85	147.39	147.92	144.54
35	5	5	56.935	37.639	52.989	51.959

According to the data obtained, the proposed PPE algorithm can find better results than the other three algorithms when the model is different, but it does not have an overwhelming advantage. For this application, the algorithm needs to be further optimized in order to achieve better results.

6 Conclusion

This paper proposes a new population-based meta-heuristic algorithm, called Phasmatodea Population Evolution algorithm. Stick insects will mimic the surrounding environment through mimicry to improve their chances of survival. As the environment changes and the population migrates, the stick insect population will gradually evolve. The algorithm in this paper mimics the evolution of stick insect populations. This paper uses 30 benchmark functions to test the proposed algorithm. Compared with PSO, GA, SA, GSA, GWO, BOA and ASO algorithms, the proposed algorithm has better competitiveness. The proposed algorithm is also tried to be applied to the downlink power allocation problem of 5G heterogeneous networks, showing a certain ability to deal with practical problems. The proposed PPE algorithm can be further optimized to improve performance and apply to more practical problems.

References

[1] I. Boussaäid, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, *Information Sciences*, Vol. 237, pp. 82-117, July, 2013.

[2] H.-S. Kim, S.-B. Cho, Application of interactive genetic algorithm to fashion design, *Engineering Applications of Artificial Intelligence*, Vol. 13, No. 6, pp. 635-644, December, 2000.

[3] F. Xhafa, A. Abraham, Computational models and heuristic methods for Grid scheduling problems, *Future Generation Computer Systems*, Vol. 26, No. 4, pp. 608-621, April, 2010.

[4] J.-S. Pan, T.-T. Nguyen, S.-C. Chu, T.-K. Dao, T.-G. Ngo, Diversity enhanced ion motion optimization for localization in wireless sensor network, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 221-229, January, 2019.

[5] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, P. Hu, Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks, *IEEE Access*, Vol. 8, pp. 8583-8594, January, 2020.

[6] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, W.-M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, Vol. 12, No. 3, Article No. 767, February, 2020.

[7] J. Zhuang, H. Luo, T.-S. Pan, J.-S. Pan, Improved flower pollination algorithm for the capacitated vehicle routing problem, *Journal of Network Intelligence*, Vol. 5, No. 3, pp. 141-156, August, 2020.

[8] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snasel, A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111-1118, July, 2018.

[9] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snasel, α-fraction first strategy for hierarchical model in wireless sensor networks, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1717-1726, November, 2018.

[10] S.-C. Chu, X. Xue, J.-S. Pan, X. Wu, Optimizing ontology alignment in vector space, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 15-22, January, 2020.

[11] J.-S. Pan, X. Wang, S.-C. Chu, T.-T. Nguyen, A multi-group grasshopper optimisation algorithm for application in capacitated vehicle routing problem, *Data Science and Pattern Recognition*, Vol. 4, No. 1, pp. 41-56, July, 2020.

[12] X.-X. Sun, J.-S. Pan, S.-C. Chu, P. Hu, A.-Q. Tian, A novel pigeon-inspired optimization with QUasi-Affine TRansformation evolutionary algorithm for DV-Hop in wireless sensor networks, *International Journal of Distributed Sensor Networks*, Vol. 16, No. 6, Article No. 1550147720932749, June, 2020.

[13] W. Hare, J. Nutini, S. Tesfamariam, A survey of non-gradient optimization methods in structural engineering, *Advances in Engineering Software*, Vol. 59, pp. 19-28, May, 2013.

[14] T. Liu, J. Wang, Z. Sun, J. Luo, T. He, K. Chen, An Improved Particle Swarm Optimization and Its Application for Micro-grid Economic Operation Optimization, in: L. Pan, G. Păun, M. J. Pérez-Jiménez, T. Song (Eds.), *Bio-Inspired Computing-Theories and Applications*, Springer, 2014, pp. 276-280.

[15] J. H. Holland, Genetic algorithms, *Scientific American*, Vol. 267, No. 1, pp. 66-73, July, 1992.

[16] J. Kennedy, R. Eberhart, Particle swarm optimization, *ICNN'95 International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[17] J.-F. Chang, S.-C. Chu, J. F. Roddick, J.-S. Pan, A Parallel Particle Swarm Optimization Algorithm with Communication Strategies, *Journal of Information Science and Engineering*, Vol. 21, No. 4, pp. 809-818, July, 2005.

[18] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 1, pp. 29-41, February, 1996.

[19] S.-C. Chu, J. F. Roddick, J.-S. Pan, Ant Colony System with Communication Strategies, *Information Sciences*, Vol. 167, No. 1, pp. 63-76, December, 2004.

[20] M. Ghosh, R. Guha, R. Sarkar, A. Abraham, A wrapper-filter feature selection technique based on ant colony optimization, *Neural Computing and Applications*, Vol. 32, No. 12, pp. 7839-7857, June, 2020.

[21] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459-471, November, 2007.

[22] P.-W. Tsai, M. K. Khan, J.-S. Pan, B. Y. Liao, Interactive Artificial Bee Colony Supported Passive Continuous Authentication System, *IEEE Systems Journal*, Vol. 8, No. 2, pp. 395-405, June, 2014.

[23] X.-S. Yang, A. H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, Vol. 29, No. 5, pp. 464-483, July, 2012.

[24] S.-J. Cai, P.-W. Tsai, Echolocation Guided Evolved Bat Algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 7, No. 1, pp. 153-162, January, 2016.

[25] X.-S. Yang, S. Deb, Cuckoo Search via Levy flights, *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, 2009, pp. 210-214.

[26] J.-S. Pan, P.-C. Song, S.-C. Chu, Y.-J. Peng, Improved Compact Cuckoo Search Algorithm Applied to Location of Drone Logistics Hub, *Mathematics*, Vol. 8, No. 3, Article No. 333, March, 2020.

[27] P.-C. Song, J.-S. Pan, S.-C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, *Applied Soft Computing*, Vol. 94, Article No. 106443, September, 2020.

[28] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[29] P. Hu, J.-S. Pan, S.-C. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, *Knowledge-Based Systems*, Vol. 195, Article No. 105746, May, 2020.

[30] J.-S. Pan, P. Hu, S.-C. Chu, Novel Parallel Heterogeneous Meta-Heuristic and Its Communication Strategies for the Prediction of Wind Power, *Processes*, Vol. 7, No. 11, Article No. 845, November, 2019.

[31] J.-S. Pan, A.-Q. Tian, S.-C. Chu, J.-B. Li, Improved binary pigeon-inspired optimization and its application for feature selection, *Applied Intelligence*, pp. 1-19, April, 2021.

[32] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, May, 1983.

[33] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Information Sciences*, Vol. 222, pp. 175-184, February, 2013.

[34] J.-S. Pan, Q.-W. Chai, S.-C. Chu, N. Wu, 3-D Terrain Node Coverage of Wireless Sensor Network Using Enhanced Black Hole Algorithm, *Sensors*, Vol. 20, No. 8, Article No. 2411, April, 2020.

[35] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Information Sciences*, Vol. 179, No. 13, pp. 2232-2248, June, 2009.

[36] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Filter modeling using gravitational search algorithm, *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 1, pp. 117-122, February, 2011.

[37] S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, Vol. 27, No. 2, pp. 495-513, February, 2016.

[38] X. Wang, J.-S. Pan, S.-C. Chu, A parallel multi-verse optimizer for application in multilevel image segmentation, *IEEE Access*, Vol. 8, pp. 32018-32030, February, 2020.

[39] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, *Acta Mechanica*, Vol. 213, No. 3-4, pp. 267-289, September, 2010.

[40] A. D'Ambrosio, D. Spiller, F. Curti, Improved magnetic charged system search optimization algorithm with application to satellite formation flying, *Engineering Applications of Artificial Intelligence*, Vol. 89, Article No.

103473, March, 2020.

[41] J.-S. Pan., X.-X. Sun, S.-C. Chu, A. Abraham, B. Yan, Digital watermarking with improved SMS applied for QR code, *Engineering Applications of Artificial Intelligence*, Vol. 97, Article No. 104049, January, 2021.

[42] G. O. Bedford, Biology and ecology of the phasmatodea, *Annual Review of Entomology*, Vol. 23, pp. 125-149, 1978.

[43] J. Skelhorn, H. M. Rowland, G. D. Ruxton, The evolution and ecology of masquerade, *Biological Journal of the Linnean Society*, Vol. 99, No. 1, pp. 1-8, January, 2010.

[44] H. Yang, C. Shi, M. S. Engel, Z. Zhao, D. Ren, T. Gao, Early specializations for mimicry and defense in a Jurassic stick insect, *National Science Review*, Vol. 8, No. 1, Article No. nwaa056, January, 2021.

[45] D. Garant, L. E. Kruuk, T. A. Wilkin, R. H. McCleery, B. C. Sheldon, Evolution driven by differential dispersal within a wild bird population, *Nature*, Vol. 433, No. 7021, pp. 60-65, January, 2005.

[46] M. T. Kinnison, N. G. Hairston, Eco-evolutionary conservation biology: contemporary evolution and the dynamics of persistence, *Functional Ecology*, Vol. 21, No. 3, pp. 444-454, June, 2007.

[47] M. A. Nowak, *Evolutionary dynamics: exploring the equations of life*, Harvard University Press, 2006.

[48] J. Liang, B. Qu, P. Suganthan, *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*, Report No. 201311, December, 2013.

[49] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Computing*, Vol. 23, No. 3, pp. 715-734, February, 2019.

[50] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Systems*, Vol. 163, pp. 283-304, January, 2019.

[51] S. Kaimaletu, R. Krishnan, S. Kalyani, N. Akhtar, B. Ramamurthi, Cognitive Interference Management in Heterogeneous Femto-Macro Cell Networks, *2011 IEEE International Conference on Communications (ICC)*, Kyoto, Japan, 2011, pp. 1-6.

[52] D. Lopez-Perez, X. Chu, A. V. Vasilakos, H. Claussen, On distributed and coordinated resource allocation for interference mitigation in self-organizing LTE networks, *IEEE/ACM Transactions on Networking*, Vol. 21, No. 4, pp. 1145-1158, August, 2013.

[53] X. Ge, J. Yang, H. Gharavi, Y. Sun, Energy efficiency challenges of 5g small cell networks, *IEEE Communications Magazine*, Vol. 55, No. 5, pp. 184-191, May, 2017.

[54] I. Ahmad, Z. Kaleem, R. Narmeen, L. D. Nguyen, D. B. Ha, Quality-of-service aware game theory-based uplink power control for 5g heterogeneous networks, *Mobile Networks and Applications*, Vol. 24, No. 2, pp. 556-563, April, 2019.

[55] K.-C. Chang, K.-C. Chu, H.-C. Wang, Y.-C. Lin, J.-S. Pan, Energy saving technology of 5G base station based on internet of things collaborative control, *IEEE Access*, Vol. 8, pp. 32935-32946, February, 2020.

Biographies

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section.

Pei-Cheng Song received his B.S. degree from Shandong University of Science and Technology, Qingdao, China, in 2019. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests are swarm intelligence and artificial intelligence.

Chun-An Pan is in his fourth year of the Bachelor of Design Computing at the University of Sydney, in the school of Architecture Design and Planning. He also started studying Bachelor of Film Production this year at SAE Qantm Creative Media Institute. He has a keen interest in researching and understanding the technological advances in the film industry and works to innovate new filmmaking technologies in the future.

Ajith Abraham received the M.Sc. degree from Nanyang Technological University, Singapore, in 1998, and the Ph.D. degree in computer science from Monash University, Melbourne, Australia, in 2001. He is currently the Director of Machine Intelligence Research Labs (MIR Labs), a Not-for-Profit Scientific Network for Innovation and Research Excellence connecting Industry and Academia.