



Simplified Phasmatodea population evolution algorithm for optimization

Pei-Cheng Song¹ · Shu-Chuan Chu¹ · Jeng-Shyang Pan¹ · Hongmei Yang¹

Received: 3 December 2020 / Accepted: 13 May 2021 / Published online: 3 June 2021
© The Author(s) 2021

Abstract

This work proposes a population evolution algorithm to deal with optimization problems based on the evolution characteristics of the Phasmatodea (stick insect) population, called the Phasmatodea population evolution algorithm (PPE). The PPE imitates the characteristics of convergent evolution, path dependence, population growth and competition in the evolution of the stick insect population in nature. The stick insect population tends to be the nearest dominant population in the evolution process, and the favorable evolution trend is more likely to be inherited by the next generation. This work combines population growth and competition models to achieve the above process. The implemented PPE has been tested and analyzed on 30 benchmark functions, and it has better performance than similar algorithms. This work uses several engineering optimization problems to test the algorithm and obtains good results.

Keywords Phasmatodea population evolution algorithm · Population competition model · Population growth model · Optimization problems

Introduction

Most of the engineering fields have complex optimization problems, which needs to be abstracted into specific models based on the existing problems, and specific methods are designed to get the optimal solution of the model [6,8,39,46]. Although there are different methods in many broader fields [5,30,48,49], including machine learning methods, data mining methods, complex network methods, etc., for problems in the optimization field, more and more different optimization algorithms are proposed [14,47,50]. Some engineering optimization problems need to obtain an accurate optimal solution. But for problems that cannot find the global opti-

mal solution in a finite time, a meta-heuristic algorithm is usually used to obtain an approximate optimal solution [17]. 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 [3,25,27,33].

After a long period of development, the meta-heuristic algorithm has been continuously developed from the classic Particle Swarm Optimization (PSO) [7,20,38], Genetic Algorithms (GA) [15], Simulated Annealing algorithm (SA) [22] and other algorithms, and there have been many other different types of algorithms applied to different problems, including Ant Colony Optimization (ACO) [10], Artificial Bee Colony algorithm(ABC) [18], Cuckoo Search algorithm(CS) [51], Gravitational Search Algorithm (GSA) [40], Butterfly Optimization Algorithm (BOA) [1], Fish Migration Optimization (FMO) [4], Grey Wolf Optimizer (GWO) [16,29], Charged System Search algorithm (CSS) [19], Atom Search Optimization (ASO) [53] and other algorithms [28]. According to different sources of inspiration, meta-heuristic algorithms can be divided into biological inspiration, physical inspiration, and evolutionary inspiration, etc. [13] Different meta-heuristic algorithms can also deal with problems in different fields and have achieved good results, such as digital watermarking [37], wireless

✉ Jeng-Shyang Pan
jspan@ieee.org
Pei-Cheng Song
spacewe@outlook.com
Shu-Chuan Chu
scchu0803@gmail.com
Hongmei Yang
yanghongmei@sdust.edu.cn

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

sensor network [11,23,26,32,34,43], vehicle routing problem [36,54], hydropower station optimization problem [44], remote sensing image contrast [45], classification of multiple power quality disturbances [52], etc.

The different algorithms mentioned above have different principles, but they are all based on the interaction between individuals to achieve different behaviors, and each iteration tries to find a better solution. The movement of an individual is mainly affected by the global optimum or the individual's historical optimum. The main information interaction is one or several good solutions that affect all other solutions [20,29]. There are fewer data interaction methods between two arbitrary individuals [1,51]. The movement of the individual is mainly achieved by approaching the global optimal or random movement, and the individual lacks certain autonomous decision-making ability [20]. In addition, the individual in the algorithm is usually directly expressed as the solution in the decision space, which corresponds to the fitness value of the solution. The individual can only rely on the fitness value or other solutions to judge the next move direction, and there is no other basis. Moreover, the way an individual calculates the direction of movement is usually a fixed formula, mostly based on the location of other individuals or random changes, and rarely chooses based on their own conditions.

Based on the above, this paper refers to the evolution characteristics of the stick insect population, and proposes the Phasmatodea population evolution algorithm (PPE) to deal with optimization problems in the N -dimensional decision space. In the following statement, stick insects have the same meaning as Phasmatodea.

Different from other algorithms that treat individuals as a solution in the decision space, this paper treats a solution as a stick insect population, and this population has the attribute of population size in addition to the corresponding calculable fitness value. The evolution of the population will be reflected in the change of the population, and the trend of population evolution will also be affected by the number of its own population. It is equivalent to providing an additional data support for the change of the solution. A population will make different decisions based on its own population size, and the change in population size is the result of all previous decisions of the population. The size of the population is also affected by environmental changes to a certain extent. An earlier version of this work was presented at the 2nd International Conference on Industrial Artificial Intelligence [42].

The PPE algorithm introduces a population competition model to realize the mutual influence of any two populations. In the real world, differences and barriers in geographic environments will affect population evolution, and there are more interactions between different populations in close and similar geographic environments. Therefore, the PPE algorithm

uses multiple nearest optimal solutions to replace the global optimal solution, increasing the diversity of population evolution trends. The algorithm mainly draws on the convergent evolution, path dependence, mutation, and population growth and competition models in the evolution of the stick insect population. The algorithm is built according to the above-mentioned characteristics, and the performance of the PPE is tested using the benchmark functions. Finally, the proposed algorithm is used to deal with some application problems, and good results are achieved.

The article is organized as follows. Section “Inspiration” introduces the related algorithms and the evolution model of the stick insect population. Section “Simplified Phasmatodea population evolution optimization algorithm” introduces the specific composition and steps of the PPE algorithm. Section “Experimental results” tests and analyzes the performance of the PPE algorithm. The PPE is proposed to deal with different application problems and demonstrated in “PPE for engineering problems”. Section “Conclusion” summarizes and discusses this paper.

Inspiration

The evolution of the stick insect population is closely related to the mimic adaptation to the environment [2]. The evolution trend of the stick insect population is to change with environmental changes and integrate itself into the surrounding environment [12,21]. Therefore, the movement of the solution in the decision space can be regarded as the evolutionary trend of the stick insect population, and the change in fitness value before and after the solution movement can be regarded as the change of the environment. The stick insect population is affected by the environmental change and will produce the next evolutionary trend.

The next evolutionary trend is affected by many characteristics. By studying the evolution process of stick insects, this article mainly chooses the following characteristics to construct the next evolutionary trend. It mainly includes convergent evolution, evolutionary path dependence, population mutation, population growth and competition models.

- (1) Convergent evolution considers the impact of the environment, and information exchange is blocked by the geographical environment.
- (2) Path dependence considers environmental changes, which will affect population evolution trends.
- (3) The population has a certain degree of initiative and will respond to changes in the environment.
- (4) Population size and competition between populations will affect population evolution.

The first is the concept of convergent evolution. For populations, similar living environments are more likely to produce similar evolutions. Due to the barrier of the geographical environment (the fitness landscape), the evolution of the population is closer to the nearest optimal solution, but not necessarily to the global optimal.

The second is the path dependence in the evolution of the population. If the population has better survival conditions after evolution, the population will generally continue the previous evolution trend. This is a relatively labor-saving evolution, but if the survival conditions become worse after the population evolves, Then the population may change the evolutionary trend.

Path dependence is similar to the greedy algorithm, which seeks the next position by continuing the previous movement trend. The underlying assumption is that the environment changes monotonously in a certain direction. The path in this article refers to the evolutionary trend of the population, and the population tends to continue the evolutionary path that achieved better results before.

The last is the population growth and competition model. In this paper, each population has two attributes: population quantity and growth rate. Population competition and changes in the environment will affect the population quantity and growth rate. When the population of a species is not restricted by the environment and resources, its population increases exponentially, but in the real environment, resources and space are limited, so the population growth model is often described by the following formula

$$\frac{dp}{dt} = rp\left(1 - \frac{p}{K}\right) \tag{1}$$

For Eq. (1), p is the population quantity, r means the effective growth rate of the population, K represents the maximum environmental capacity of the population of a species.

In the PPE algorithm, this paper uses the logistic difference equation, let $K = 1$ [31]. Then, using the population quantity at t can calculate the population quantity at $t + 1$ as

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

For Eq. (2), a represents the growth rate, p is between 0 and 1, and the value of a is between 0 and 4. When a exceeds the interval 0–4, 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 and fall into chaos.

In addition to the impact of environment and resources on population growth, competition between different pop-

ulations will also affect the number and growth rate of populations, and the disadvantaged populations may disappear. The population competition model is usually described by the following equation:

$$\frac{dp}{dt} = r_1p \left(1 - \frac{p}{n_1} - s_1 \frac{q}{n_2}\right) \tag{3}$$

In Eq. (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 .

Based on the above description of the evolution characteristics of the stick insect population, combined with the population growth and competition model, this paper attempts to implement the PPE algorithm and improve its performance.

Simplified Phasmatodea population evolution optimization algorithm

In this paper, an improved stick insect population evolution algorithm is designed to deal with the minimization of n -dimensional space. This section attempts to design a new heuristic optimization algorithm, trying to integrate historical population decision data, population autonomous decision-making ability, and interaction between arbitrary populations into the algorithm framework.

The algorithm mainly includes three parts: initialization, population position update, and population evolution trend update. Next, the three parts are introduced separately.

Initialization

Suppose there are Np solutions in the n -dimensional space, and each solution is represented by x , and x is a point in the n -dimensional space, that is, $x = [x_1, x_2, \dots, x_n]$. We regard a solution x as a stick insect population, assuming that it has certain autonomous decision-making ability. Each population has two attributes: population quantity and growth rate, $p_i(1 \leq i \leq Np)$ represents the quantity of the i -th population (solution), and a_i means the growth rate of the i -th population.

First, randomly generate Np populations (solutions) x , and the population quantity of each population x is calculated by

$$p_i = \frac{1}{Np} \tag{4}$$

For later use, the population quantity p_i of the i -th population is initialized to a decimal in Eq. (4). In addition, the growth rate a of each population is set to a fixed value of 1.1, which means that all populations have a good survival state in the initial situation, and then as the environment changes (the population position x changes), the quantity of populations will change.

Based on the introduction in “Inspiration”, for population x , its evolution process is more similar to the dominant population with similar living environment and closer distance, and it is not necessarily able to achieve convergent evolution to the global optimal population. This is not only due to the barrier of the geographical environment, but also because the characteristics of the global optimal population cannot make the population x find a better position in the current area.

Therefore, this article selects k historical optimal solutions to guide the movement of the surrounding solutions, stored in Ho , $Ho = [x_{h1}, \dots, x_{hi}, \dots, x_{hk}]$. 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

$$k = \lceil \log(Np) \rceil + 1 \tag{5}$$

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.

Ho is used to store the best first k solutions in the past search process, and the judgment of the quality of the solution is obtained by calculating the fitness value. Ho is used to store the nearest optimal solution to guide the movement of other solutions around.

Population position update

After the population is initialized, the population position needs to be updated. After the update of the population position x^t at time t , it will move to a new position $x^t + 1$ and in a new environment, this position change is represented by the evolution trend ev . The position update formula is

$$x^{t+1} = x^t + ev \tag{6}$$

For Eq. (6), ev is the evolutionary trend of stick insect populations. After the population position changes, it is necessary to re-evaluate all populations, update the global optimal solution $gbest$ and k optimal solutions Ho .

Population evolution trend

The update of the population trend mainly includes three parts. The first two parts choose different ways to update

according to the comparison of the quality before and after the trend update. The third part uses the population competition model to update. The first two parts represent the choice of the autonomous decision-making strategy of the population, there are two kinds, and the third part represents the mutual influence between the population.

The first is the update of the first two parts, which mainly use the characteristics of population growth model, convergent evolution and path dependence. After the population position is updated, the updated population is evaluated. If it is better than before, then the first part of the population trend update is performed, otherwise the second part of the population trend update method is performed.

The first part first updates the population quantity, and the update equation is

$$p^{t+1} = a^{t+1} p^t (1 - p^t) \tag{7}$$

After using Eq. (7) to update p , the population trend ev needs to be updated. The update method mainly contains three parts.

The first is the convergence to the nearest optimal, and secondly, because the previous evolutionary trend ev^t has obtained better results, then the next update is more likely to continue the previous evolutionary trend, which is called path dependence. Path dependence is similar to the greedy algorithm, which seeks the next position by continuing the previous movement trend. The underlying assumption is that the environment changes monotonously in a certain direction. Finally, there are mutations within the population. Therefore, the update equation of population trend is as follows:

$$ev^{t+1} = (1 - p^{t+1})A + p^{t+1}(ev^t + m) \tag{8}$$

For Eq. (8), A represents the degree of approach to the nearest optimal, and the calculation formula is

$$A = (s(Ho, x^t) - x^t) \cdot c \tag{9}$$

For Eq. (9), $s(Ho, x^t)$ is used to find the historical optimal solution that is closest to x^t in Ho , c is the coefficient of the influence degree of the nearest optimal solution on the current population, usually less than 1, which is a fixed value of 0.2 in this article. m represents the mutation of the population in some dimensions, first randomly select w dimension ($w \leq n$), then use the standard normal distribution to generate a random number. Finally, an n -dimensional mutation vector m is formed by $m = m(Ub - Lb) \times 0.2$, Lb is the lower limit and Ub is the upper limit of n -dimensional space, $Lb = [lb_1, \dots, lb_n]$, $Ub = [ub_1, \dots, ub_n]$.

The second part is that the population position does not obtain a better value after the previous update, then the pop-

ulation will no longer maintain the original trend, but tend to the nearest optimal solution, and at the same time produce unpredictable disturbances. However, before this, the population has a certain probability to maintain the relatively poor situation after the update, so as to have the opportunity to jump out of the current area. In the process, the population size will change according to Eq. (7).

Based on the above description, the probability that the population accepts a poor solution is set to p_i , that is, the quantity of current population. And the population trend update formula in the second part is

$$ev^{t+1} = \text{rand} \cdot A + st \cdot B \tag{10}$$

For Eq. (10), rand represents an n -dimensional random vector generated using a uniform distribution, and each dimension is between 0 and 1, and B represents an n -dimensional random vector generated using a standard normal distribution. st is initially set to $(Ub - Lb) \times 0.1$, and with each iteration of the algorithm, st is updated to $st = st \times 0.99$.

The third part is the impact of the competition between populations on the evolution trend of the population. The population competition model refers to Eq. (3), and a simple modification is made here. The revised formula is

$$p_i = p_i + a_i p_i \left(1 - p_i - \frac{f(x_j)}{f(x_i)} p_j \right) \tag{11}$$

When the population x_i is competing with x_j , the population quantity of x_i is calculated using Eq. (11). Compared to Eq. (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 .

Where x_i is the current population, x_j is randomly selected from other $Np - 1$ populations, and then the distance between the two populations needs to be judged and compared with a given threshold G . The calculation formula for G is

$$G = 0.1 \times (Ub - Lb) \frac{Max_gen + 1 - t}{Max_gen} \tag{12}$$

For Eq. (12), Max_gen means the number of iterations, and t represents the current number of iterations. The threshold G will become smaller and smaller as the number of iterations increases.

If the distance between the current population x_i and other randomly selected population x_j is less than the threshold G , then it is considered that the two populations have competed. Competition will have an impact on the evolution trend of the current population x_i , which is described by the following formula

$$ev^{t+1} = ev^{t+1} + \frac{f(x_j) - f(x_i)}{f(x_j)} (x_j - x_i) \tag{13}$$

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 population is generated to replace, indicating that the current stick insect population disappears due to poor adaptability to the environment.

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

Algorithm 1: PPE

```

Initialize  $Np$  populations;
Initialize  $ev, p, k$  use Eq 4, Eq 5;
Calculate fitness  $f(x)$ , set  $gbest$  and  $Ho$ ;
for  $t = 2$  to  $Max\_gen$  do
    Update each  $x$  to  $newx$  use Eq 6;
    Calculate new fitness  $f(newx)$ , update  $gbest$  and  $Ho$ ;
    for  $i = 1$  to  $Np$  do
        if  $f(newx) \leq f(x)$  then
            Update  $x = newx$ , update  $f(x)$ ;
            Update  $p_i$  use Eq 7;
            Update  $ev_i$  use Eq 8 and 9;
        else
            if  $rd < p_i$  then
                Update  $x = newx$ , update  $f(x)$ ;
                Update  $p_i$  use Eq 7;
            Update  $ev_i$  use Eq 10;
    Randomly choose a solution  $x_j, (j \neq i)$ ;
    if  $dist(x_j, x_i) < G$  then
        Update  $p_i$  use Eq 11, update  $ev_i$  use Eq 13;
    
```

In Algorithm 1, rd means a random number generated using a uniform distribution, and the range is 0–1. 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>).

Experimental results

Benchmark function and parameter settings

Based on the previous introduction, this section tests the performance of the proposed PPE. This article uses benchmark functions to test the performance of the PPE and analyzes the performance of the PPE algorithm in different dimensions.

There are many test functions for related algorithms, but there are some common test suites in the CEC competition. This paper uses CEC2014 [24] 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).

To show the performance of the proposed PPE algorithm more objectively and truthfully, the more classic PSO [20]

Table 1 Parameters setting of every different algorithm

Algorithm	Parameter
PSO	$N_p = 20$, $c_1 = 2$, $c_2 = 2$, inertia constant: 0.8–0.2 linearly decreasing
SLPSO	$N_p = 20$ ($m = 20$), $M = 100$, $\alpha = 0.5$, $\beta = 0.01$
GA	$N_p = 20$, crossover rate = 0.4, mutation rate = 0.1
SA	$N_p = 1$, initial temp = 0.1, temp reduction rate = 0.99, mutation rate = 0.5
GSA	$N_p = 20$, initial gravitational constant = 100, decreasing coefficient = 20
GWO	$N_p = 20$
BOA	$N_p = 20$, power exponent = 0.1, modular modality = 0.01, probability switch = 0.8
ASO	$N_p = 20$, depth weight = 50, multiplier weight = 0.2
PPE	$N_p = 20$

and GA [15] algorithms are selected in the manuscript, and then several other algorithms proposed in recent years are selected, including the GSA algorithm in 2009 [40] and the GWO algorithm in 2014 [29], Social Learning Particle Swarm Optimization (SLPSO) algorithm in 2015 [7], BOA [1] and ASO [53] algorithms in 2019. The parameter settings of related algorithms are shown in Table 1.

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

Comparison with other algorithms

After determining the algorithm to be compared with and the parameters of the corresponding algorithm, for a fairer comparison, it is also necessary to ensure that different algorithms perform the same function evaluation times. Therefore, the function evaluation times of all algorithms in this article are 40,000. The dimension is 30, each different algorithm is run 30 times on different functions and the average value is recorded. The overall performance of each different algorithm compared with the PPE was measured at a significant level $\alpha = 0.05$ under the Wilcoxon's sign rank test. The results are shown in Table 4.

Table 2 is the test results of the proposed PPE and other algorithms on 30 benchmark functions. The symbol (<) means that the algorithm does not perform as well as the PPE on the current benchmark function. And the symbol (=) means that the two algorithms perform similarly on the current benchmark function. And the symbol (>) means that the PPE performs poorly. In the last row of the table, the comparison results on all functions are summarized.

Shown in Table 4 are the Wilcoxon's sign rank test results of each algorithm compared with the PPE algorithm at a significant level of 0.05. The symbol (+) indicates that the

performance of the algorithm under the current benchmark function is significantly worse than that of the PPE algorithm, the symbol (–) indicates that the performance of the current algorithm is stronger than the PPE algorithm, and the symbol (\approx) indicates that there is no significant difference between the performance of the current algorithm and the PPE algorithm.

For Table 2, the overall performance of the PPE on the 30 functions of CEC2014 is better than SLPSO, GA, SA, BOA, GSA, GWO algorithms, and similar to the overall performance of PSO and ASO algorithms. Compared with SA algorithm, the performance of PPE on 25 benchmark functions is better than SA algorithm. Compared with the PSO, the PPE only has better performance on 16 benchmark functions than the PSO, and cannot achieve an overwhelming advantage. At the same time, the performance of the PPE is better than that of the ASO on only 15 benchmark functions. Due to the randomness of the algorithm test, the PPE cannot achieve an overwhelming advantage.

To compare the performance of different algorithms on different types of benchmark functions more clearly, this paper sorts out the data in Table 2, finds out which algorithm obtains the optimal solution for each benchmark function, and summarizes according to the type of benchmark function. The results are shown in Table 3.

Table 3 shows and compares the performance of the algorithm on different types of benchmark functions.

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, the PPE has achieved the best solution among all algorithms on seven functions, and at the same time, stronger than all other algorithms. The BOA algorithm performs well on the composition function, which is stronger than other algorithms, but the effect is worse than other algorithms on other types of functions.

Convergence evaluation

For meta-heuristic algorithms, it is not only necessary to compare the final results of different algorithms, but also to consider the convergence of the algorithm. Algorithms with different convergence speeds may eventually obtain similar results, so this article shows the convergence of the proposed PPE algorithm, as shown in Figs. 1 and 2.

Due to space limitations, only partial results are shown here. The type of the selected functions are different, the performance of the algorithms are also different, and all the test results are displayed on the website (<https://github.com/Spacewe-outlook/PPE>).

In Figs. 1 and 2, the horizontal axis is the number of function evaluations. To show the difference between different curves more clearly, the vertical axis is $\text{Log}(f - f^*)$. f

Table 2 Comparison of average of fitness functions on 30D optimization among PSO, GA, SA, GSA, GWO, BOA and ASO

$F(x)$	PSO	SLPSO	GA	SA	BOA	GSA	GWO	ASO	PPE
F1	1.42E+07 (<)	8.27E+07 (<)	2.04E+07 (<)	4.26E+07 (<)	1.52E+09 (<)	2.82E+07 (<)	1.02E+08 (<)	1.85E+07 (<)	1.11E+07 (<)
F2	7.61E+03 (>)	7.78E+08 (<)	5.27E+07 (<)	2.02E+09 (<)	7.39E+10 (<)	9.42E+03 (<)	4.49E+09 (<)	9.43E+03 (<)	2.41E+05 (>)
F3	5.20E+03 (<)	6.80E+04 (<)	2.13E+04 (<)	7.61E+03 (<)	8.02E+04 (<)	5.99E+04 (<)	4.38E+04 (<)	7.67E+03 (<)	3.12E+03 (<)
F4	6.15E+02 (<)	6.56E+02 (<)	5.74E+02 (<)	6.89E+02 (<)	1.72E+04 (<)	7.67E+02 (<)	8.06E+02 (<)	5.96E+02 (<)	5.38E+02 (<)
F5	5.21E+02 (<)	5.20E+02 (<)	5.20E+02 (<)	5.21E+02 (<)	5.21E+02 (<)	5.20E+02 (<)	5.21E+02 (<)	5.20E+02 (<)	5.20E+02 (>)
F6	6.15E+02 (>)	6.18E+02 (>)	6.22E+02 (>)	6.26E+02 (>)	6.39E+02 (<)	6.22E+02 (<)	6.17E+02 (>)	6.09E+02 (>)	6.25E+02 (>)
F7	7.00E+02 (>)	7.08E+02 (<)	7.01E+02 (<)	7.19E+02 (<)	1.52E+03 (<)	7.00E+02 (<)	7.26E+02 (<)	7.00E+02 (>)	7.00E+02 (>)
F8	8.36E+02 (>)	8.85E+02 (>)	8.20E+02 (>)	9.34E+02 (>)	1.13E+03 (<)	9.49E+02 (<)	9.05E+02 (<)	8.76E+02 (>)	9.02E+02 (>)
F9	9.76E+02 (>)	9.99E+02 (>)	1.03E+03 (>)	1.14E+03 (>)	1.25E+03 (<)	1.07E+03 (<)	1.02E+03 (>)	9.59E+02 (>)	1.06E+03 (>)
F10	2.31E+03 (>)	3.67E+03 (<)	1.86E+03 (<)	5.19E+03 (<)	8.79E+03 (<)	4.61E+03 (<)	3.66E+03 (<)	3.82E+03 (<)	3.01E+03 (<)
F11	5.43E+03 (<)	4.88E+03 (>)	4.61E+03 (>)	7.49E+03 (>)	9.01E+03 (<)	5.19E+03 (<)	4.23E+03 (>)	4.37E+03 (>)	4.91E+03 (>)
F12	1.20E+03 (<)	1.20E+03 (>)	1.20E+03 (>)	1.20E+03 (<)	1.20E+03 (<)	1.20E+03 (>)	1.20E+03 (<)	1.20E+03 (>)	1.20E+03 (>)
F13	1.30E+03 (<)	1.30E+03 (<)	1.30E+03 (<)	1.30E+03 (<)	1.31E+03 (<)	1.30E+03 (<)	1.30E+03 (<)	1.30E+03 (<)	1.30E+03 (<)
F14	1.40E+03 (<)	1.40E+03 (<)	1.40E+03 (<)	1.40E+03 (<)	1.71E+03 (<)	1.40E+03 (<)	1.41E+03 (<)	1.40E+03 (<)	1.40E+03 (<)
F15	1.51E+03 (>)	1.99E+03 (<)	1.53E+03 (<)	1.55E+03 (<)	4.29E+05 (<)	1.51E+03 (<)	1.82E+03 (<)	1.50E+03 (>)	1.52E+03 (>)
F16	1.61E+03 (>)	1.61E+03 (<)	1.61E+03 (<)	1.61E+03 (>)	1.61E+03 (<)	1.61E+03 (<)	1.61E+03 (>)	1.61E+03 (>)	1.61E+03 (>)
F17	1.26E+06 (<)	4.05E+06 (<)	3.11E+06 (<)	1.85E+06 (<)	1.71E+08 (<)	1.39E+06 (<)	3.64E+06 (<)	4.64E+05 (<)	4.62E+05 (<)
F18	1.05E+06 (<)	4.55E+03 (<)	8.21E+03 (<)	8.33E+05 (<)	5.34E+09 (<)	2.40E+03 (<)	1.42E+07 (<)	3.10E+03 (<)	3.04E+03 (<)
F19	1.91E+03 (>)	1.94E+03 (<)	1.93E+03 (<)	1.92E+03 (>)	2.48E+03 (>)	2.00E+03 (<)	1.95E+03 (<)	1.96E+03 (<)	1.92E+03 (<)
F20	8.86E+03 (<)	4.29E+04 (<)	4.32E+04 (<)	6.15E+03 (<)	4.60E+05 (<)	5.23E+04 (<)	3.78E+04 (<)	1.93E+04 (<)	7.69E+03 (<)
F21	3.37E+05 (<)	2.51E+06 (<)	7.16E+05 (<)	8.87E+05 (<)	1.01E+08 (<)	2.69E+05 (<)	1.96E+06 (<)	4.05E+05 (<)	2.13E+05 (<)
F22	2.55E+03 (>)	2.81E+03 (>)	2.90E+03 (>)	2.76E+03 (<)	6.48E+04 (>)	3.14E+03 (<)	2.65E+03 (>)	2.73E+03 (>)	2.86E+03 (>)
F23	2.62E+03 (<)	2.63E+03 (<)	2.62E+03 (<)	2.63E+03 (<)	2.50E+03 (<)	2.64E+03 (<)	2.64E+03 (<)	2.62E+03 (<)	2.62E+03 (<)
F24	2.64E+03 (<)	2.66E+03 (<)	2.63E+03 (<)	2.66E+03 (<)	2.60E+03 (<)	2.61E+03 (>)	2.60E+03 (>)	2.63E+03 (<)	2.63E+03 (<)
F25	2.71E+03 (>)	2.72E+03 (<)	2.72E+03 (<)	2.71E+03 (>)	2.70E+03 (>)	2.71E+03 (>)	2.71E+03 (>)	2.71E+03 (>)	2.71E+03 (>)
F26	2.74E+03 (>)	2.77E+03 (>)	2.76E+03 (>)	2.79E+03 (<)	2.77E+03 (<)	2.80E+03 (<)	2.75E+03 (>)	2.70E+03 (>)	2.78E+03 (>)
F27	3.37E+03 (<)	3.49E+03 (<)	3.45E+03 (<)	3.44E+03 (<)	3.59E+03 (<)	4.68E+03 (<)	3.39E+03 (<)	3.18E+03 (<)	3.28E+03 (>)
F28	4.24E+03 (>)	4.45E+03 (>)	4.40E+03 (>)	4.43E+03 (>)	5.83E+03 (>)	6.96E+03 (>)	4.28E+03 (<)	5.53E+03 (>)	6.79E+03 (>)
F29	2.68E+06 (<)	9.23E+05 (<)	4.50E+05 (<)	1.03E+06 (<)	1.03E+06 (<)	1.16E+05 (<)	1.25E+06 (<)	6.66E+07 (<)	1.03E+04 (<)
F30	9.49E+03 (>)	4.12E+04 (<)	1.11E+04 (<)	1.17E+04 (<)	3.20E+03 (<)	6.44E+04 (<)	5.77E+04 (<)	3.48E+04 (<)	1.05E+04 (<)
< / = / >	16/0/14	22/0/8	22/0/8	25/0/5	23/0/7	21/0/9	21/0/9	15/0/15	-

Table 3 Performance of different algorithms on different types of functions

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

represents the final result obtained by the algorithm on a benchmark function. f^* represents the minimum value of the current benchmark function.

In Fig. 1, the final results of the PPE algorithm on the benchmark functions F1 and F3 are better than other algorithms, but in the early stage of algorithm execution, the PPE algorithm's convergence speed is slower than other algorithms. The final results of the PPE algorithm on the benchmark functions F5 and F10 are worse than other algorithms, but in the early stage of the algorithm execution, the convergence speed of the algorithm is better than that of the PSO, GWO, and GSA algorithms.

According to Fig. 2, the proposed PPE algorithm has continuous convergence ability on the benchmark functions F17 and F20. In the later stage of the algorithm execution, the proposed PPE algorithm is not easy to fall into a local optimum compared to other algorithms, but the convergence speed is slower. The PPE algorithm has better performance on the benchmark function F21, and the final result is slightly better than all other algorithms, but the performance on the benchmark function F30 is more general.

Compared with other algorithms, the PPE algorithm has a downward trend within the given evaluation times. Compared with other algorithms, most of them no longer continue to converge, and the convergence curve tends to be stable in the later stage. It can be seen that it cannot find a better result and falls into a local optimum.

Since the number of the nearest optimal solutions in the PPE algorithm is calculated by Eq. (5), when Np is equal to 20, k is 3.

In the case of keeping Np equal to 20, this paper tests whether the number of the nearest optimal solutions will affect the convergence speed of the algorithm and how it will affect. This article selected several different results, and the test results are shown in Fig. 3.

It can be seen from Fig. 3 that when k is different, the convergence curve of the PPE algorithm is also different. For the benchmark function F2, when k is smaller, the algorithm converges faster and the result is the best. As k increases, the performance of the algorithm keeps getting worse. For the benchmark functions F6 and F14, the smaller the k , the faster the algorithm converges, but the final results of different k

are closer. But for the benchmark function F19, the smaller the k , the faster the algorithm converges, but the final result is not as good as when k is larger. It can be seen that when k is relatively small, it is helpful for the rapid convergence of the algorithm, but the diversity of the solution cannot be fully guaranteed, and it is still possible to fall into a local optimum.

Scalability analysis

Since the above test mainly uses a benchmark function with a dimension of 30 for testing, in practical applications, the dimension of the optimization problem is not necessarily 30. Therefore, to show the performance of the PPE algorithm more comprehensively, it is necessary to display and analyze the performance of the algorithm in different dimensions. The comparison results are shown in Figs. 4 and 5.

According to Fig. 4, the performance of the PPE algorithm on the 10-, 30-, 50- and 100-dimensional benchmark function F1 is better than other algorithms. On the F3, the PPE algorithm can obtain better results when the dimension is relatively low, but when the dimension is 100, the algorithm performance is worse than the GA and ASO algorithms. The performance of the algorithm PPE on the benchmark functions F6 and F7 is worse than the ASO and PSO algorithms.

According to Fig. 5, the PPE algorithm benchmark functions F14, F17, F18, and F29 have achieved good results, but they do not have advantages in every dimension. On the benchmark function F14, the PPE algorithm has a poor effect when the dimension is 10. But the effect is better when the dimension becomes higher. The PPE algorithm has certain advantages in different dimensions on the benchmark function F29, but it is worse than the BOA algorithm.

Visualization of algorithm changes

To show the internal changes of the PPE algorithm more clearly, this paper selects the benchmark functions F1 and F8 to visualize the internal changes of the PPE algorithm during the execution of the algorithm. The results are shown in Figs. 6 and 7.

Table 4 Compared with the PPE, the results of each algorithm is measured at a significant level $\alpha = 0.05$ under the Wilcoxon's signed rank test

$F(x)$	PSO	SLPSO	GA	SA	GSA	GWO	BOA	ASO
F1	5.13E-01 (≈)	3.13E-08 (+)	7.49E-04 (+)	3.13E-08 (+)	2.65E-04 (+)	8.41E-08 (+)	3.13E-08 (+)	3.79E-01 (≈)
F2	3.13E-08 (-)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (-)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (-)
F3	3.79E-01 (≈)	3.13E-08 (+)	8.49E-06 (+)	1.06E-06 (+)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (+)	4.47E-03 (+)
F4	6.70E-06 (+)	1.51E-05 (+)	1.57E-02 (+)	3.61E-08 (+)	3.13E-08 (+)	7.31E-08 (+)	3.13E-08 (+)	4.47E-03 (+)
F5	3.13E-08 (+)	2.40E-04 (+)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (-)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (-)
F6	2.51E-07 (-)	3.68E-06 (-)	1.37E-02 (-)	7.63E-01 (-)	1.97E-03 (-)	2.51E-07 (-)	3.13E-08 (+)	3.13E-08 (-)
F7	3.13E-08 (-)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (+)	4.25E-07 (-)	3.13E-08 (+)	3.13E-08 (+)	2.41E-08 (-)
F8	3.13E-08 (-)	1.69E-02 (-)	3.13E-08 (-)	2.13E-05 (+)	5.56E-07 (+)	8.41E-01 (≈)	3.13E-08 (+)	2.40E-04 (-)
F9	1.92E-07 (-)	1.37E-06 (-)	1.53E-03 (-)	1.06E-06 (+)	2.27E-01 (≈)	6.23E-04 (-)	3.13E-08 (+)	3.13E-08 (-)
F10	4.69E-06 (-)	1.97E-04 (+)	4.16E-08 (-)	3.13E-08 (+)	1.46E-07 (+)	9.83E-04 (+)	3.13E-08 (+)	6.34E-05 (+)
F11	4.35E-01 (≈)	1.00E+00 (≈)	1.38E-01 (≈)	3.13E-08 (+)	1.25E-01 (≈)	3.82E-03 (-)	3.13E-08 (+)	1.19E-02 (-)
F12	3.13E-08 (+)	1.57E-02 (-)	6.84E-04 (-)	3.13E-08 (+)	3.13E-08 (-)	1.40E-03 (+)	3.13E-08 (+)	3.13E-08 (-)
F13	2.15E-03 (+)	1.06E-06 (+)	3.90E-04 (+)	3.13E-08 (+)	6.69E-01 (≈)	1.20E-05 (+)	3.13E-08 (+)	9.00E-01 (≈)
F14	2.38E-05 (+)	2.25E-06 (+)	1.99E-06 (+)	3.13E-08 (+)	8.26E-02 (≈)	3.61E-08 (+)	3.13E-08 (+)	6.23E-04 (+)
F15	3.75E-07 (-)	1.07E-05 (-)	1.69E-02 (+)	9.66E-08 (+)	7.31E-08 (-)	1.70E-05 (+)	3.13E-08 (+)	3.13E-08 (-)
F16	2.06E-02 (-)	3.27E-01 (≈)	1.31E-04 (-)	7.06E-01 (≈)	4.88E-07 (+)	1.17E-03 (-)	1.07E-05 (+)	2.68E-01 (≈)
F17	7.49E-04 (+)	1.20E-05 (+)	7.31E-08 (+)	8.20E-07 (+)	6.84E-04 (+)	2.55E-06 (+)	3.13E-08 (+)	1.00E+00 (≈)
F18	9.65E-05 (+)	1.25E-01 (≈)	1.07E-04 (+)	3.13E-08 (+)	9.69E-02 (≈)	3.61E-08 (+)	3.13E-08 (+)	9.00E-01 (≈)
F19	1.20E-05 (-)	2.34E-03 (+)	3.14E-01 (≈)	1.19E-04 (-)	6.33E-07 (+)	1.90E-05 (+)	3.13E-08 (+)	4.42E-02 (+)
F20	9.60E-01 (≈)	9.66E-08 (+)	1.06E-06 (+)	2.36E-02 (-)	3.13E-08 (+)	2.87E-07 (+)	3.13E-08 (+)	1.76E-06 (+)
F21	2.47E-01 (≈)	1.66E-03 (+)	1.78E-04 (+)	8.70E-05 (+)	2.27E-01 (≈)	4.12E-05 (+)	3.13E-08 (+)	3.65E-01 (≈)
F22	3.26E-06 (-)	5.97E-01 (≈)	6.51E-01 (≈)	1.38E-01 (≈)	3.54E-04 (+)	1.17E-03 (-)	3.13E-08 (+)	5.27E-02 (≈)
F23	1.67E-07 (+)	3.13E-08 (+)	1.25E-01 (+)	3.13E-08 (+)	3.13E-08 (+)	3.13E-08 (+)	3.31E-09 (-)	8.41E-08 (+)
F24	2.25E-06 (+)	3.13E-08 (+)	1.99E-06 (+)	3.13E-08 (+)	5.28E-06 (-)	3.13E-08 (-)	3.31E-09 (-)	3.52E-01 (≈)
F25	2.15E-03 (-)	2.17E-04 (+)	6.27E-02 (≈)	1.82E-01 (≈)	1.51E-05 (-)	2.91E-01 (≈)	3.31E-09 (-)	1.47E-02 (-)
F26	1.25E-01 (≈)	2.06E-02 (-)	2.68E-01 (≈)	3.46E-02 (+)	1.97E-03 (+)	1.93E-02 (-)	4.90E-04 (-)	4.15E-06 (-)
F27	1.80E-02 (+)	3.52E-03 (+)	6.59E-03 (+)	2.76E-03 (+)	7.31E-08 (+)	1.11E-02 (+)	4.71E-04 (+)	1.52E-01 (≈)
F28	3.61E-08 (-)	3.13E-08 (-)	8.41E-08 (-)	3.13E-08 (-)	6.51E-01 (≈)	3.13E-08 (-)	6.23E-04 (-)	3.70E-05 (-)
F29	1.35E-05 (+)	1.07E-03 (+)	3.70E-05 (+)	4.27E-07 (+)	9.60E-01 (≈)	1.27E-07 (+)	3.31E-09 (-)	3.68E-02 (+)
F30	1.07E-01 (≈)	4.16E-08 (+)	7.44E-01 (≈)	9.69E-02 (≈)	4.16E-08 (+)	3.13E-08 (+)	3.31E-09 (-)	9.57E-03 (+)

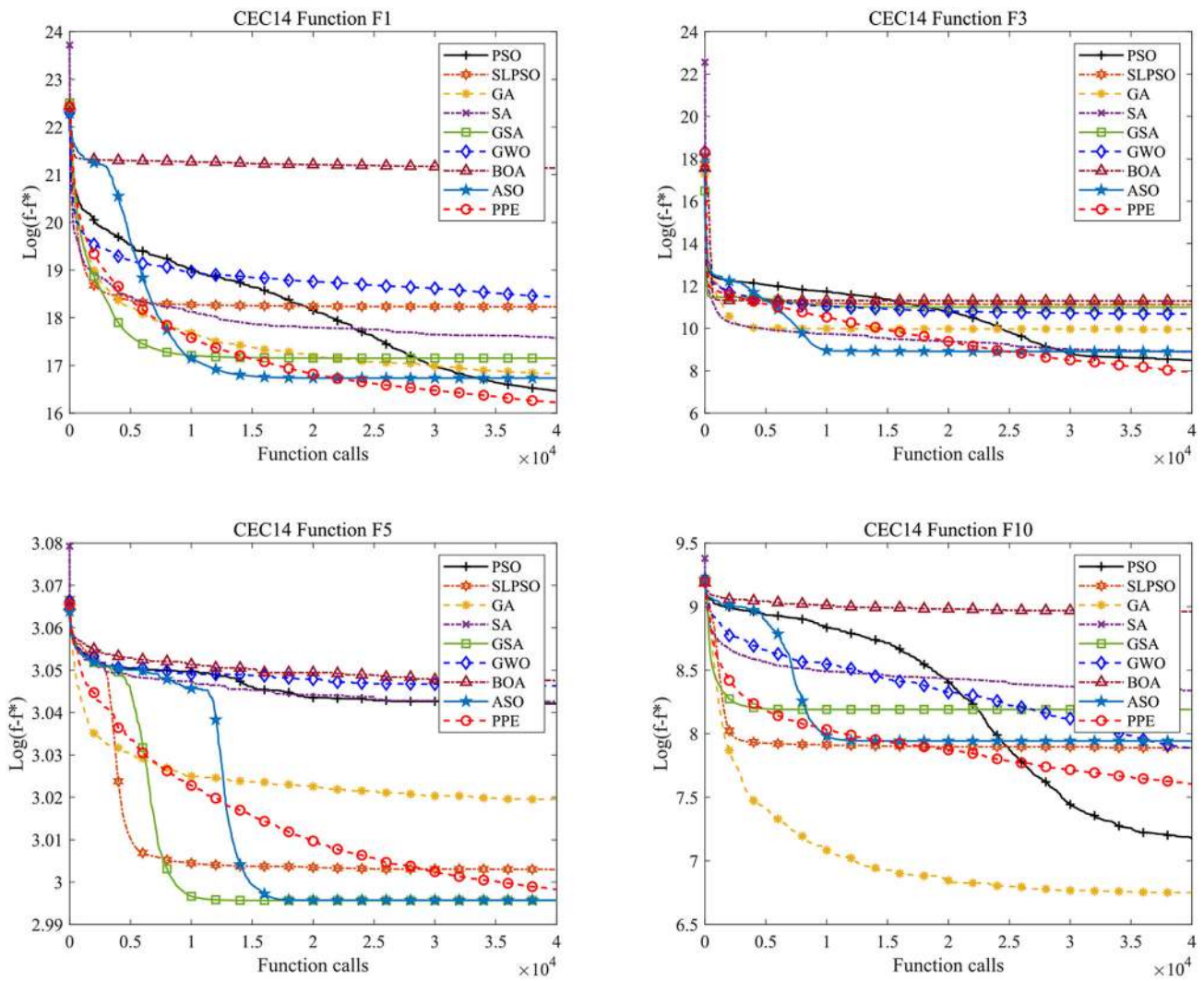


Fig. 1 Convergence test results in 30 dimensions (F1, F3, F5, F10)

Figure 6 shows the characteristics of different aspects of the algorithm PPE in the process of iterating 1000 times on the benchmark function F1.

The first picture is a three-dimensional map of the function F1 in two dimensions.

The second figure shows the changes in the best fitness value and the worst fitness value obtained in all populations throughout the iteration process. It can be seen that the best fitness value of Np populations has been changing, but the relative change is relatively small, and the worst fitness value gradually becomes better with relatively small changes, but the fluctuation of each iteration process is large.

The third picture shows the change process of the population number of the first population in the iteration process. It can be seen that the change of the population number p is irregular.

According to Algorithm 1, since the PPE algorithm will first judge whether a better solution can be obtained after the population position is updated, if a better solution is obtained, the first part will be executed, otherwise the second part will be executed. Therefore, the fourth picture compares the ratio of the number of times the first part and the second part are executed to the total number of times in the entire iteration process. Therefore, the fourth picture compares the ratio of the number of times the first part and second part is executed to the total number of times.

It can be seen from the fourth figure that in the six test results, the ratio of PPE algorithm obtaining better results on the benchmark function F1 is higher than the ratio of obtaining worse results.

Figure 7 shows the internal changes of the PPE algorithm when iterating on the benchmark function F8. Different from Fig. 6, according to the fourth picture of Fig. 7, the ratio of the

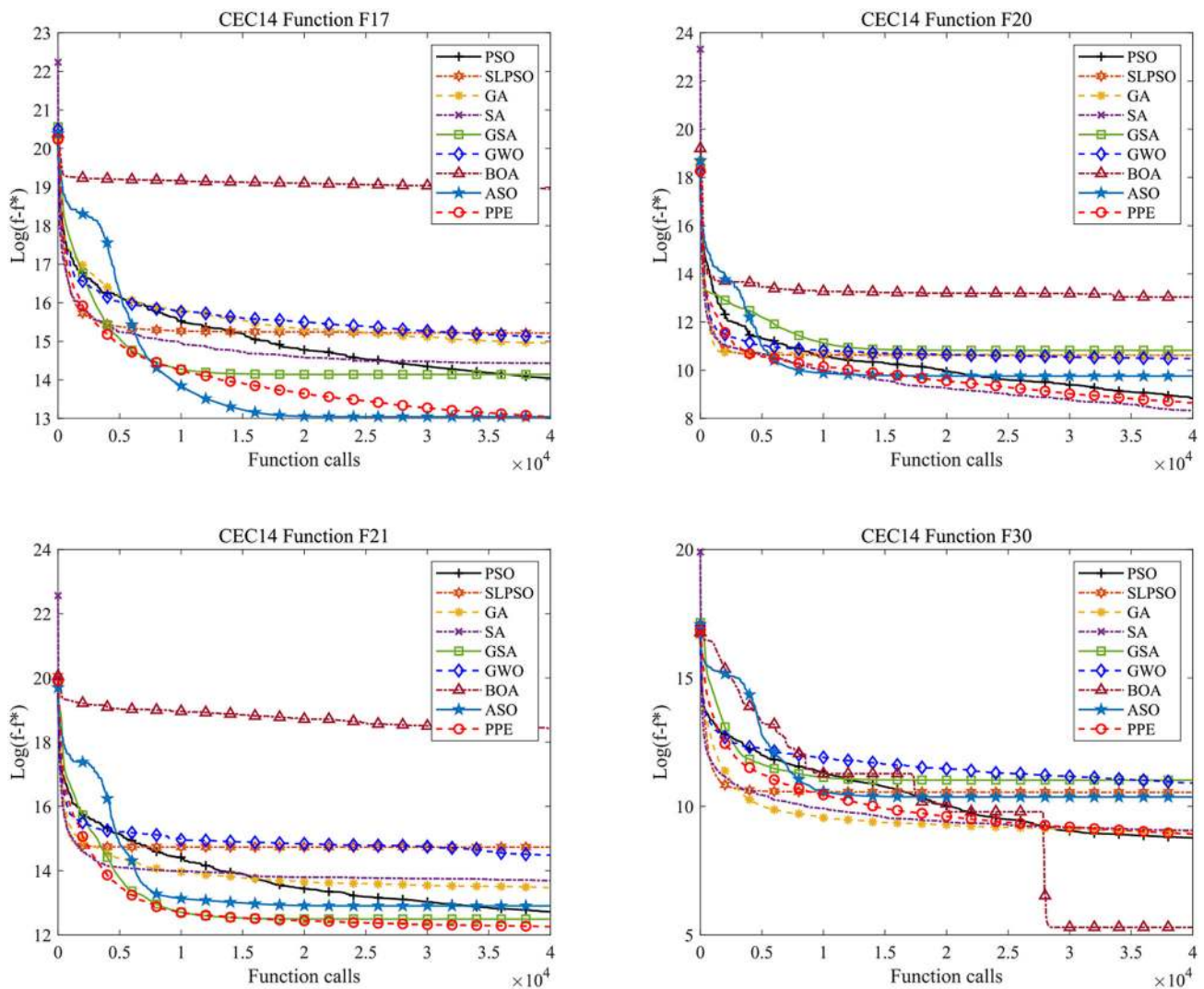


Fig. 2 Convergence test results in 30 dimensions (F17, F20, F21, F30)

PPE algorithm to obtain a worse position in the whole iterative process is higher than that of a better position. According to the second picture, the worst fitness values obtained by Np populations have not gradually improved and the fluctuations are relatively stable, and the best fitness values have also converged slowly.

PPE for engineering problems

This section tests the application of the proposed PPE algorithm on engineering optimization problems, and selects several well-known engineering problems, including welded beam design, compression spring design and pressure vessel design and drone logistics hub location.

The parameter settings of all algorithms and the number of generated solutions are shown in Table 1, and are not mod-

ified in this section. The evaluation times of all algorithms in different engineering problems are 40,000. All algorithms are run five times in different problems, recording the average, best value, standard deviation and worst value. At the same time, the paper gives the rank-sum test results of the best values obtained by different algorithms.

Welded beam design

Welding beam design problem is a basic and classic engineering optimization problem, often used to test the performance of optimization algorithms. This problem is a minimization problem, which minimizes the manufacturing cost of the welded beam [9].

Four variables need to be considered when optimizing this problem, including thickness of weld (h), height of the bar (t), length of the clamped bar (l) and thickness of the bar (b).

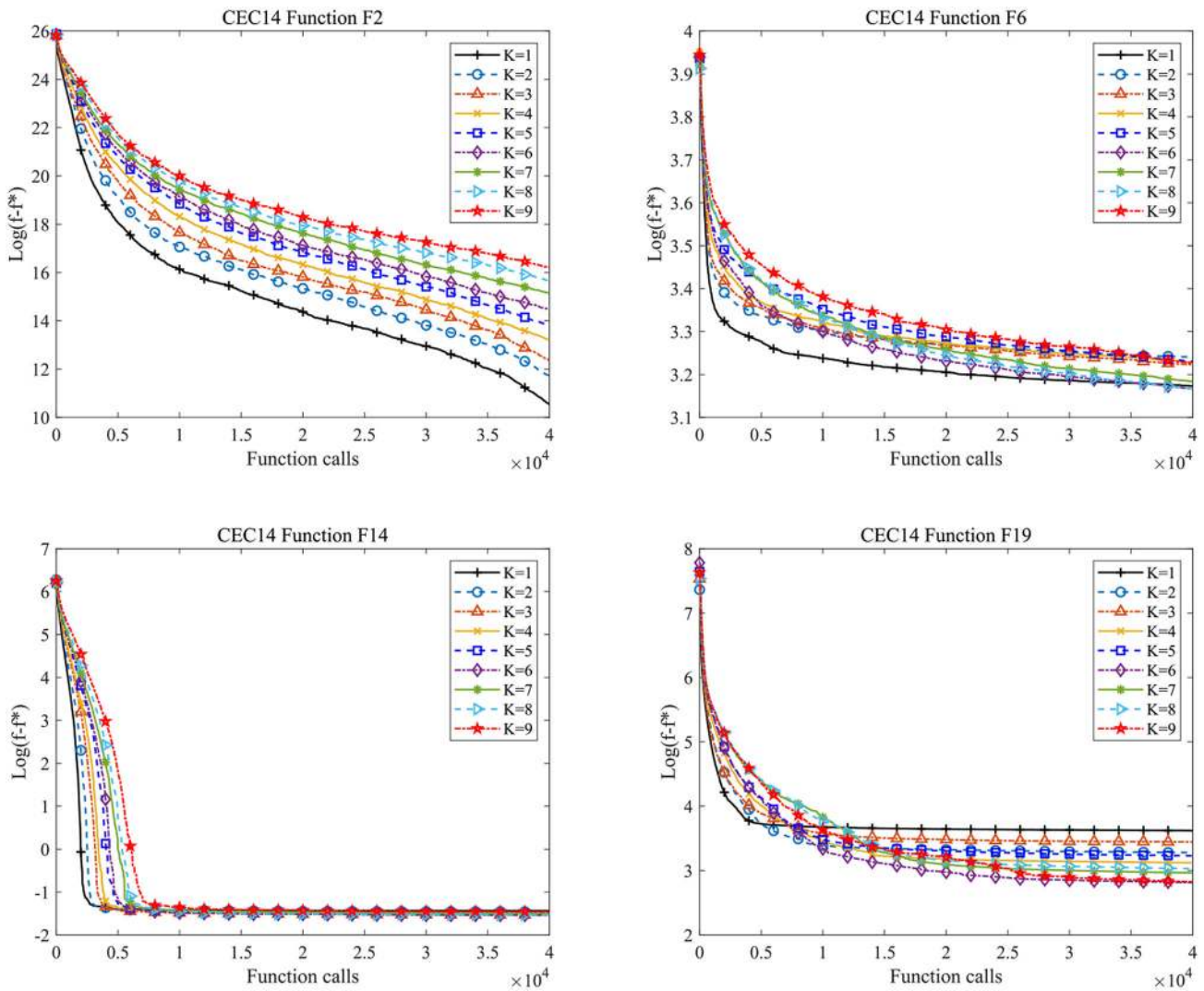


Fig. 3 Convergence test result when k is different (F2, F6, F14, F19)

In addition to this, multiple constraints need to be considered, the mathematical representation is as follows

$$0.1 \leq g_3 \leq 10, \\ 0.1 \leq g_4 \leq 2$$

Consider $\mathbf{g} = [g_1, g_2, g_3, g_4] = [h, l, t, b]$

$$\begin{aligned} \min \quad & f(\mathbf{g}) = g_1^2 g_2 1.10471 \\ & + (14.0 + g_2) g_3 g_4 0.04811 \\ \text{st.} \quad & y_1(\mathbf{g}) = \tau(\mathbf{g}) - \tau_{\max} \leq 0 \\ & y_2(\mathbf{g}) = \sigma(\mathbf{g}) - \sigma_{\max} \leq 0 \\ & y_3(\mathbf{g}) = \delta(\mathbf{g}) - \delta_{\max} \leq 0 \\ & y_4(\mathbf{g}) = g_1 - g_4 \leq 0 \\ & y_5(\mathbf{g}) = P - P_c(\mathbf{g}) \leq 0 \\ & y_6(\mathbf{g}) = 0.125 - g_1 \leq 0 \\ & y_7(\mathbf{g}) = 1.10471 g_1^2 + 0.0481 g_3 g_4 \\ & \quad \times (14.0 + g_2) - 5.0 \leq 0 \end{aligned}$$

Variable constraint interval $0.1 \leq g_1 \leq 2,$
 $0.1 \leq g_2 \leq 10,$

where

$$\begin{aligned} \tau(\mathbf{l}) &= \sqrt{\tau'^2 + 2\tau'\tau''(g_2/R) + (\tau'')^2} \\ \tau' &= P/\sqrt{2g_1g_2}, \tau'' = MR/J, M = p(L + g_2/2) \\ R &= \sqrt{\frac{g_2^2}{4} + (g_1 + g_3/2)^2} \\ J &= 2 \left\{ \sqrt{2}g_1g_2 \left[\left(\frac{g_2^2}{4} \right) + \left(\frac{g_1 + g_3}{2} \right)^2 \right] \right\} \\ P_c(\mathbf{l}) &= \frac{4.013E\sqrt{\frac{g_3^2g_4^6}{36}}}{L^2} \left(1 - \frac{g_3}{2L}\sqrt{E/4G} \right) \\ \sigma_{\max} &= 30000\text{psi}, P = 6000\text{lb}, L = 14\text{in.}, \end{aligned}$$

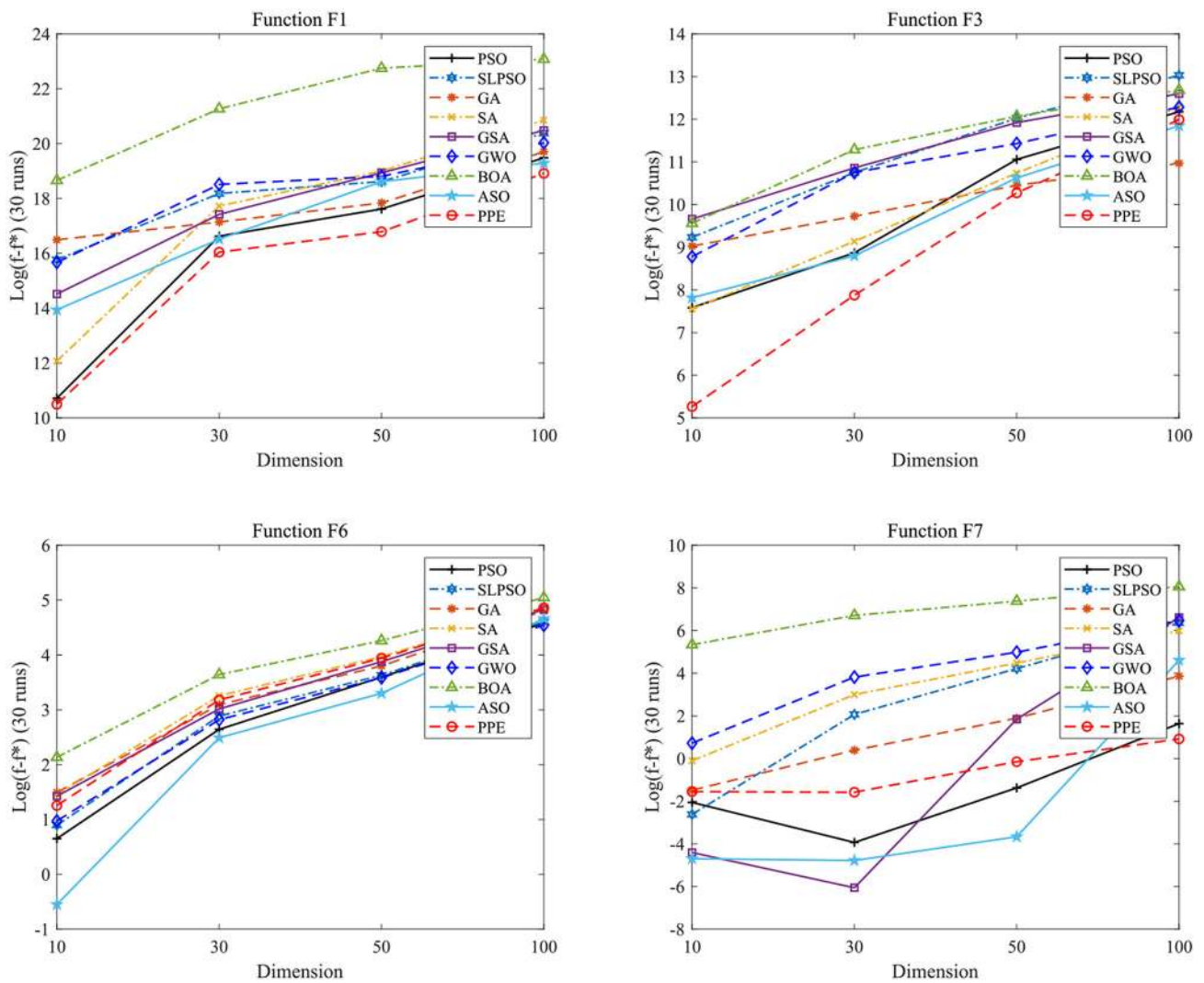


Fig. 4 Comparison of different algorithms with different dimensions in different functions (F1, F3, F6, F7)

$$\delta_{\max} = 0.25in., E = 3 \times 10^6 \text{psi}, \tau_{\max} = 13600 \text{psi}$$

$$G = 12 \times 10^6 \text{psi}$$

Based on the above description, this paper uses different algorithms to test, the results are shown in Table 5.

Compression spring design

This problem is also a minimization engineering problem often used to test optimization algorithms. This problem requires minimizing the weight of tension and compression springs [41]. The three variables to be considered in the problem optimization process are the diameter of the wire (d), mean coil diameter (D) and the number of the active coils (N). In addition to this, multiple constraints need to be con-

sidered, the mathematical representation of the problem is as follows

Consider $\mathbf{g} = [g_1, g_2, g_3] = [d, D, N]$

$$\min f(\mathbf{g}) = (g_3 + 2)g_2g_1^2$$

$$\text{st. } y_1(\mathbf{g}) = 1 - \frac{g_2^3g_3}{71785g_1^4} \leq 0,$$

$$y_2(\mathbf{g}) = \frac{4g_2^2 - g_1g_2}{12566(g_2g_1^3 - g_1^4)} + \frac{1}{5108g_1^2} \leq 0,$$

$$y_3(\mathbf{g}) = 1 - \frac{140.45g_1}{g_2^2g_3} \leq 0,$$

$$y_4(\mathbf{g}) = \frac{g_1 + g_2}{1.5} - 1 \leq 0$$

Variable range $0.05 \leq g_1 \leq 2.00, 0.25 \leq g_2 \leq 1.30,$
 $2.00 \leq g_3 \leq 15.00$

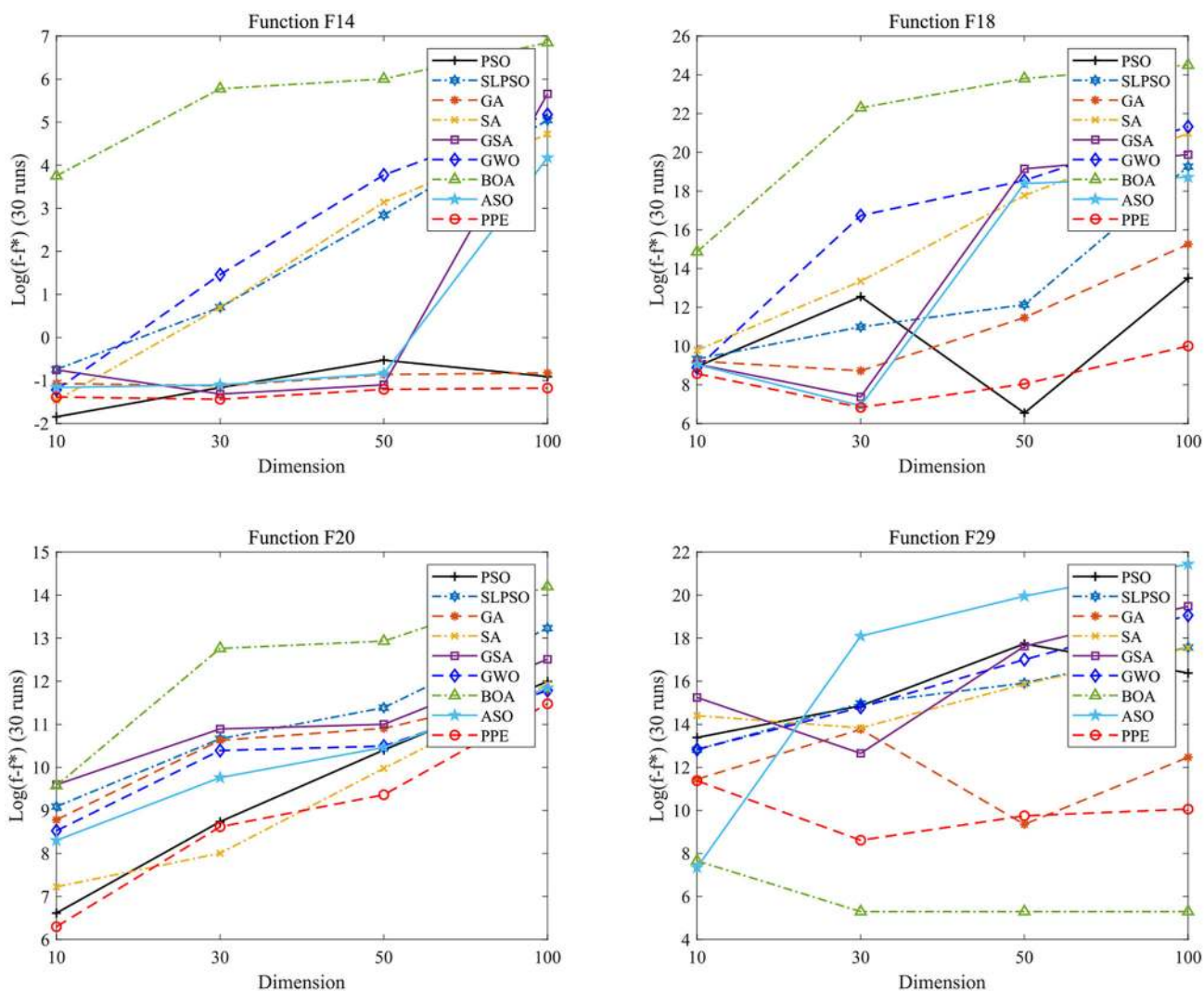


Fig. 5 Comparison of different algorithms with different dimensions in different functions (F14, F18, F20, F29)

Combined with the above description of the problem, this paper uses related algorithms to test the problem, the results are shown in Table 6. Compared with other algorithms, the best value obtained by the PPE algorithm is better than other algorithms.

Pressure vessel design

The goal of the pressure vessel design problem is to minimize the total cost of the cylindrical pressure vessel. Variables to consider include the shell thickness (T_s), the thickness of the head (T_h), the inner radius (R), the section length of the vessel (L). In addition to this, multiple constraints need to be considered, the mathematical representation is as follows

$$\begin{aligned} \text{Consider } \mathbf{g} &= [g_1, g_2, g_3, g_4] \\ &= [T_s, T_h, R, L] \end{aligned}$$

$$\begin{aligned} \min \quad f(\mathbf{g}) &= 0.6224g_1g_3g_4 + 1.7781g_2g_3^2 \\ &\quad + 3.1661g_1^2g_4 + 19.84g_1^2g_3 \\ \text{st. } \quad y_1(\mathbf{g}) &= -g_1 + 0.0193g_3 \leq 0, \\ y_2(\mathbf{g}) &= -g_3 + 0.00954g_3 \leq 0, \\ y_3(\mathbf{g}) &= -\pi g_3^2g_4 - \frac{4}{3}\pi g_3^3 \\ &\quad + 1296000 \leq 0, \\ y_4(\mathbf{g}) &= g_4 - 240 \leq 0 \end{aligned}$$

Variable constraint interval

$$\begin{aligned} 0 &\leq g_1 \leq 99, \\ 0 &\leq g_2 \leq 99, \\ 10 &\leq g_3 \leq 200, \\ 10 &\leq g_4 \leq 200 \end{aligned}$$

This article uses the engineering problem to test the PPE and other algorithms, the results are shown in Table 7. Compared with other algorithms, the PPE algorithm can obtain

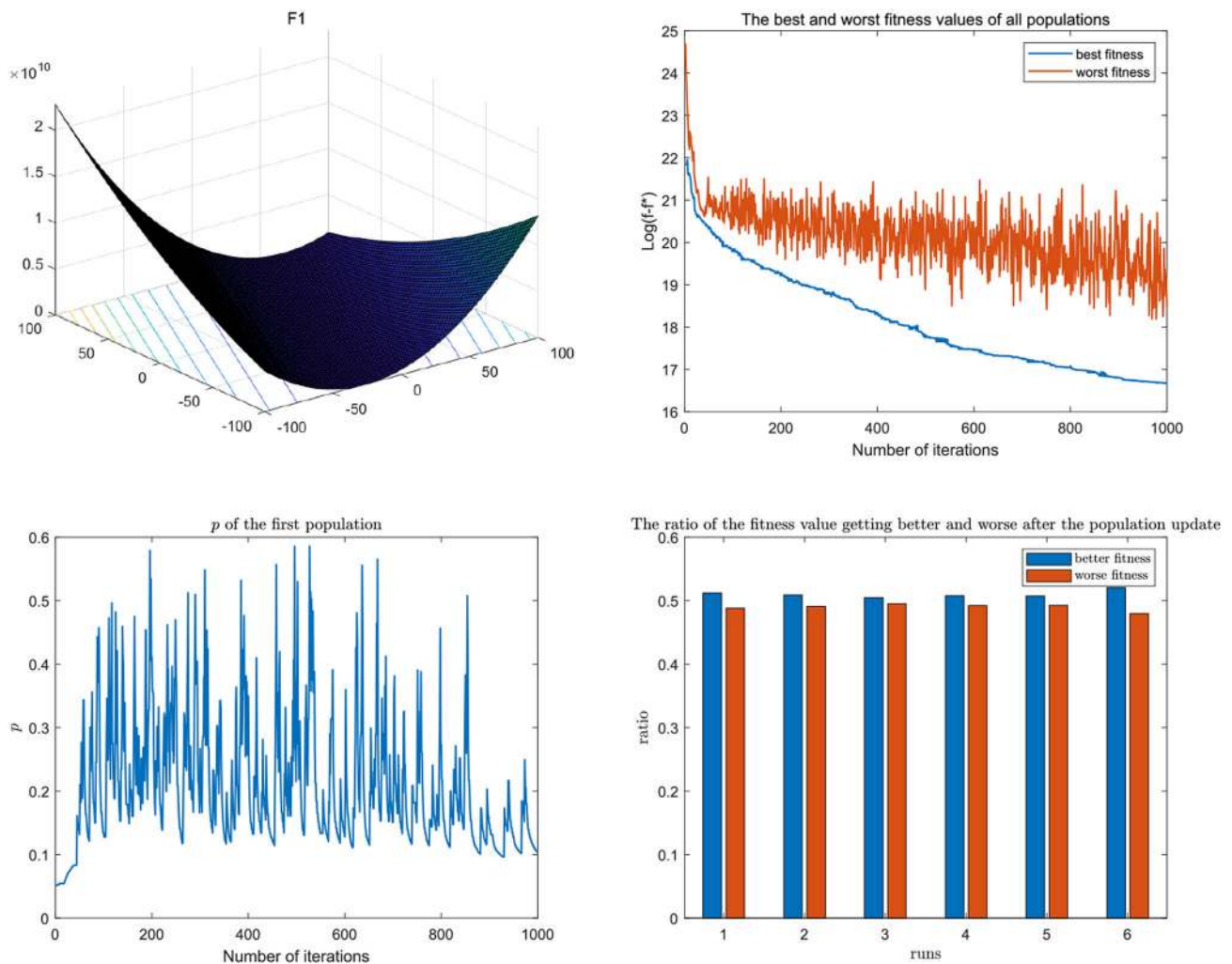


Fig. 6 Analysis of the algorithm execution process in F1

better results, but it is not as stable as the GWO algorithm. Since the algorithm is not specifically designed for the above application problems, the performance gap between PPE algorithm and GWO, ASO and other algorithms is not obvious. Based on the above comparison results, although the proposed PPE algorithm does not have an overwhelming advantage on a specific problem, it still has similar performance compared with existing algorithms.

Drone logistics hub location

This paper uses the Drone logistics hub location model proposed in [35] to test the proposed PPE algorithm.

This model considers the situation where the drone logistics hub serves the surrounding villages. Each village is in charge of a logistics hub, and each logistics hub is responsible for multiple villages. The location of the drone logistics hub is selected considering the location of the surrounding villages and the population size and the difficulty of transportation

from the current location to the village. The objective function constructed by the model is

$$\min F = \sum_{k=1, k \in N} (H_k - R_k) \cdot cp_k \cdot \left(\frac{L1_k}{L2_k} \right), \tag{14}$$

where N means the number of villages, k means the k -th village, cp_k is represented as the population of the k -th village, R_k means the radius of a village, and H_k represents the straight-line distance from a village to the nearest logistics hub. $L2_k$ is the same as H_k , and $L1_k$ represents the land distance from the current village to the nearest logistics hub. This paper randomly generates several models. The two-dimensional area simulated by the model is $50,000 \text{ m} \times 50,000 \text{ m}$. The village size and population are randomly generated. The radius of the village ranges from 200 to 900, and the village population ranges from 300 to 3000.

Table 8 is the test results of different algorithms when using different models. Table 8 is the test results of different

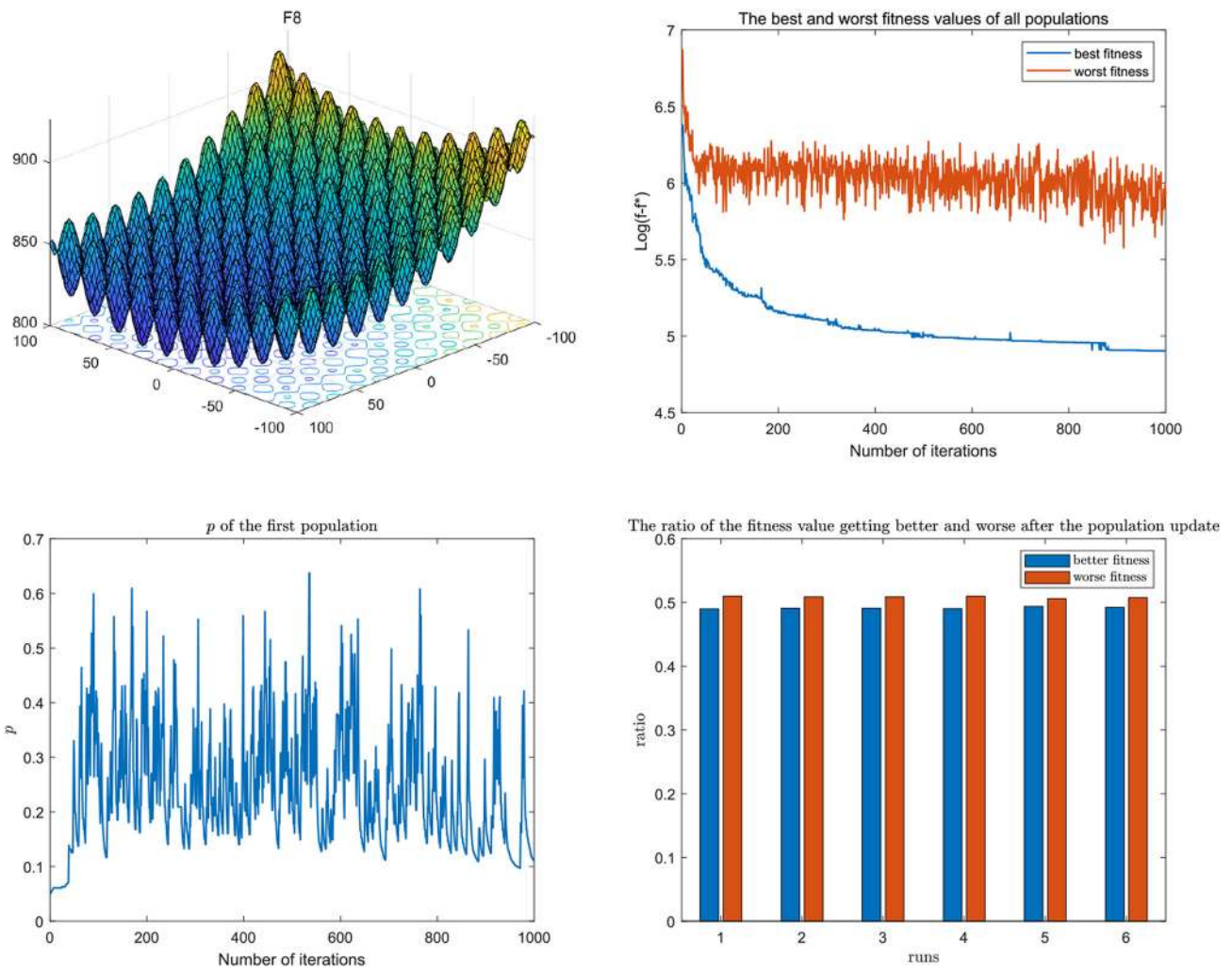


Fig. 7 Analysis of the algorithm execution process in F8

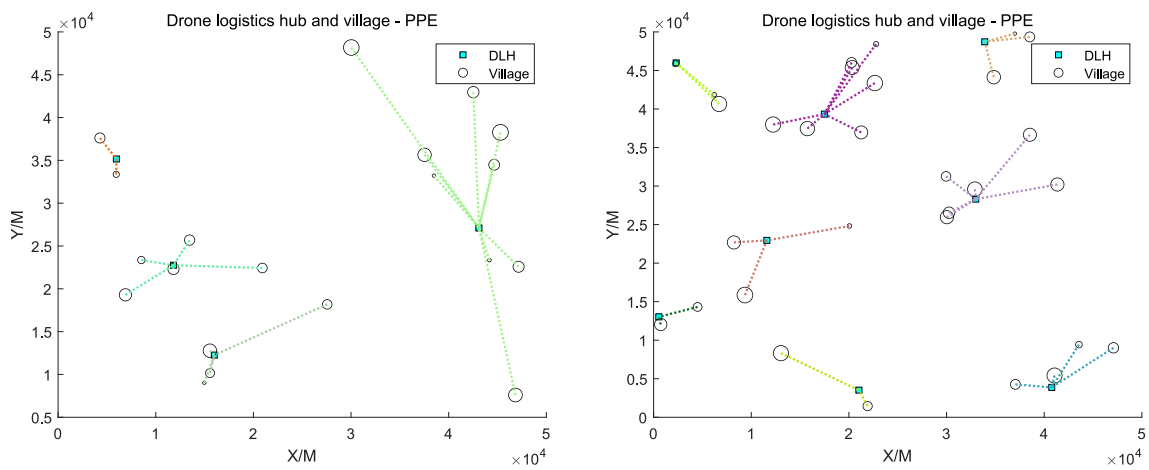


Fig. 8 PPE algorithm test results on Model 1 ($D = 4, N = 20$) and Model 2 ($D = 8, N = 30$)

Table 5 Results of different algorithms for welded beam design problem

Algorithm	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	Best	Mean	Std	Worst	rank-sum(Best)
PSO	0.205734	3.253034	9.036624	0.20573	1.695245	1.695245	2.49E−08	1.695245	0.007937
GA	0.300352	2.600315	6.964762	0.34635	2.185657	2.652367	0.40979	3.116103	0.007937
GSA	2.050336	2.050336	2.050336	2.050336	2.050336	2.33331	0.172526	2.484519	0.015873
GWO	0.205331	3.261676	9.037218	0.205729	1.695921	1.697652	0.001629	1.699825	0.690476
BOA	0.175872	4.775395	7.16808	0.363982	2.519892	3.413816	0.58865	4.080176	0.007937
ASO	0.137484	5.081883	9.036624	0.20573	1.812825	1.89355	0.100337	2.01839	0.150794
PPE	0.205722	3.253798	9.035708	0.205777	1.695528	1.770569	0.166447	2.068318	–

Table 6 Results of different algorithms for compression spring design problem

Algorithm	<i>d</i>	<i>D</i>	<i>N</i>	Best	Mean	Std	Worst	rank-sum(Best)
PSO	0.139131	1.299956	11.89605	3.661977	3.662078	9.13E−05	3.662223	0.007937
GA	0.13412	1.163302	14.32213	3.723919	4.368244	0.841057	5.750541	0.007937
GSA	3.674108	3.674108	3.674108	3.674108	4.360719	1.132282	6.331865	0.007937
GWO	0.139154	1.3	11.89367	3.661894	3.661896	1.66E−06	3.661899	0.015873
BOA	0.093774	0.822113	9.914266	13.8017	39.08341	33.43898	89.13724	0.007937
ASO	0.138787	1.286613	12.13466	3.667751	3.690985	0.021613	3.713126	0.007937
PPE	0.139149	1.3	11.89239	3.661893	3.661894	8.82E−07	3.661896	–

Table 7 Results of different algorithms for pressure vessel design problem

Algorithm	<i>T_s</i>	<i>T_h</i>	<i>R</i>	<i>L</i>	Best	Mean	Std	Worst	rank-sum(Best)
PSO	0.964351	0.022456	63.43148	39.60738	3630.935	4626.963	777.1485	5695.286	0.007937
GA	0.790107	0	52.51519	79.56467	3361.681	3484.05	86.83775	3577.926	0.007937
GSA	3588.093	3588.093	3588.093	3588.093	3588.093	6708.446	5900.806	17227.85	0.007937
GWO	1.093367	0	65.22536	10	2302.559	3053.841	1678.047	6055.622	0.150794
BOA	0.922636	0	66.24475	10	2792.724	7020.761	5341.769	16060.67	0.055556
ASO	0.539805	2.1E−11	42.97202	175.6732	3785.204	5326.184	2494.587	9756.325	0.007937
PPE	1.112585	2.37E−05	65.24628	10.02023	2309.667	2854.043	484.4381	3220.821	–

Table 8 The results of the PPE algorithm and other algorithms on the location of the drone logistics hub in Model 1 and 2

M	D	N	PPE	PSO	GSA	BOA
Model 1	4	20	3.717E+08	3.741E+08	6.623E+08	5.163E+08
Model 1	5	20	2.935E+08	2.996E+08	5.208E+08	4.378E+08
Model 1	7	20	2.097E+08	2.111E+08	4.447E+08	3.602E+08
Model 1	9	20	1.920E+08	2.015E+08	3.889E+08	3.330E+08
Model 2	6	30	5.050E+08	5.424E+08	7.934E+08	7.497E+08
Model 2	8	30	3.632E+08	3.741E+08	6.871E+08	6.966E+08

algorithms when using different models, M means the number of models, D means the number of drone hubs, and N means the number of villages generated. And the number of iterations is 1000.

In Table 8, the algorithm can get better results than other algorithms in the two generated models, and the generated results are shown in Fig. 8.

Conclusion

By imitating the evolution law of stick insect population, this paper proposes and implements a population evolution algorithm for optimization problems. The proposed PPE algorithm realizes convergent evolution, path dependence and intra-population mutation, and combines population growth and competition models. In this paper, the performance of the proposed PPE is tested, compared with classic and newer similar algorithms, and the algorithm effect is tested on several engineering problems. The results show that the PPE algorithm has better performance, and It can be further optimized. Since the PPE algorithm is not specifically applied to a certain type of practical problems, it is necessary to improve the algorithm for practical application problems in the next work to get better performance and achieve more practical results.

Acknowledgements An earlier version of this work was presented at the 2nd International Conference on Industrial Artificial Intelligence.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23(3):715–734. <https://doi.org/10.1007/s00500-018-3102-4>
- Bedford GO (1978) Biology and ecology of the phasmatodea. *Annu Rev Entomol* 23(1):125–149
- Boussaid I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. *Inf Sci* 237:82–117
- Chai QW, Chu SC, Pan JS, Zheng WM (2020) Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-D terrain. *J Inf Hiding Multimed Signal Process* 11(2):90–102
- Chen CM, Chen L, Gan W, Qiu L, Ding W (2021) Discovering high utility-occupancy patterns from uncertain data. *Inf Sci* 546:1208–1229. <https://doi.org/10.1016/j.ins.2020.10.001>
- Chen YC, Syamsudin M, Xu W (2019) An internet of things thermostat sensor developed with an arduino device using a recursively digital optimization algorithm. *J Inf Hiding Multimed Signal Process* 10(3):434–446
- Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
- Chu SC, Xue X, Pan JS, Wu X (2020) Optimizing ontology alignment in vector space. *J Int Technol* 21(1):15–22
- Coello Coello CA (2000) Constraint-handling using an evolutionary multiobjective optimization technique. *Civ Eng Syst* 17(4):319–346
- Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 26(1):29–41
- Du ZG, Pan JS, Chu SC, Luo HJ, Hu P (2020) Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks. *IEEE Access* 8:8583–8594
- Garant D, Kruuk LE, Wilkin TA, McCleery RH, Sheldon BC (2005) Evolution driven by differential dispersal within a wild bird population. *Nature* 433(7021):60–65
- Hare W, Nutini J, Tesfamariam S (2013) A survey of non-gradient optimization methods in structural engineering. *Adv Eng Softw* 59:19–28
- He C, Huang S, Cheng R, Tan KC, Jin Y (2020) Evolutionary multiobjective optimization driven by generative adversarial networks (GANs). *IEEE Trans Cybern* 1–14. <https://doi.org/10.1109/TCYB.2020.2985081>
- Holland JH (1992) Genetic algorithms. *Sci Am* 267(1):66–73
- Hu P, Pan JS, Chu SC (2020) Improved binary grey wolf optimizer and its application for feature selection. *Knowl Based Syst* 195. <https://doi.org/10.1016/j.knsys.2020.105746>
- Hu X, Eberhart RC, Shi Y (2003) Engineering optimization with particle swarm. In: Proceedings of the 2003 IEEE swarm intelligence symposium. SIS'03 (Cat. No.03EX706), pp 53–57. <https://doi.org/10.1109/SIS.2003.1202247>
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Optim* 39(3):459–471
- Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213(3–4):267–289
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, IEEE, vol 4, pp 1942–1948
- Kinnison MT, Hairston NG Jr (2007) Eco-evolutionary conservation biology: contemporary evolution and the dynamics of persistence. *Funct Ecol* 21(3):444–454
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Kong L, Pan JS, Tsai PW, Vaclav S, Ho JH (2015) A balanced power consumption algorithm based on enhanced parallel cat swarm optimization for wireless sensor network. *Int J Distrib Sensor Netw* 11(3)
- Liang J, Qu B, Suganthan P (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, Computational Intelligence Laboratory, p 635
- Liao P, Sun C, Zhang G, Jin Y (2020) Multi-surrogate multi-tasking optimization of expensive problems. *Knowl Based Syst* 205. <https://doi.org/10.1016/j.knsys.2020.106262>
- Liu N, Pan JS, Wang J et al (2019) An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks. *Sensors* 19(19):4112
- Liu N, Pan JS, Sun C, Chu SC (2020) An efficient surrogate-assisted quasi-affine transformation evolutionary algorithm for expensive

- optimization problems. *Knowl Based Syst* 209. <https://doi.org/10.1016/j.knosys.2020.106418>
28. Meng Z, Pan JS, Tseng KK (2019) PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization. *Knowl-Based Syst* 168:80–99
 29. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
 30. Nguyen TT, Dao TK, Horng MF, Shieh CS (2016) An energy-based cluster head selection algorithm to support long-lifetime in wireless sensor networks. *J Netw Intell* 1(1):23–37
 31. Nowak MA (2006) *Evolutionary dynamics: exploring the equations of life*. Harvard University Press, USA
 32. Pan JS, Meng Z, Chu SC, Xu HR (2017) Monkey king evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment. *Telecommun Syst* 65(3):351–364
 33. Pan JS, Kong L, Sung TW, Tsai PW, Snášel V (2018) A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set. *J Internet Technol* 19(4):1111–1118
 34. Pan JS, Nguyen TT, Chu SC, Dao TK, Ngo TG (2019) Diversity enhanced ion motion optimization for localization in wireless sensor network. *J Inf Hiding Multimed Signal Process* 10(1):221–229
 35. Pan JS, Song PC, Chu SC, Peng YJ (2020a) Improved compact cuckoo search algorithm applied to location of drone logistics hub. *Mathematics* 8(3):333
 36. Pan JS, Wang X, Chu SC, Nguyen TT (2020b) A multi-group grasshopper optimisation algorithm for application in capacitated vehicle routing problem. *Data Sci Patt Recogn* 4(1):41–56
 37. Pan JS, Sun XX, Chu SC, Abraham A, Yan B (2021) Digital watermarking with improved SMS applied for QR code. *Eng Appl Artif Intell* 97. <https://doi.org/10.1016/j.engappai.2020.104049>
 38. Qin S, Sun C, Zhang G, He X, Tan Y (2020) A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems. *Complex Intell Syst* 6(2):263–274
 39. Rao SS (2019) *Engineering optimization: theory and practice*. Wiley, Hoboken
 40. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
 41. Rather SA, Bala PS (2020) Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems. *World J Eng*
 42. Song PC, Chu SC, Pan JS, Yang H (2020) Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine. In: 2020 2nd international conference on industrial artificial intelligence (IAI), IEEE, pp 1–5. <https://doi.org/10.1109/IAI50351.2020.9262236>
 43. Sun XX, Pan JS, Chu SC, Hu P, Tian AQ (2020) A novel pigeon-inspired optimization with QUasi-Affine TRansformation evolutionary algorithm for DV-Hop in wireless sensor networks. *Int J Distrib Sens Netw* 16(6):1550147720932749
 44. Tian AQ, Chu SC, Pan JS, Cui H, Zheng WM (2020) A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station. *Sustainability* 12(3):767
 45. Tu TN, Hoang HN, Van TV, Van DD, Phuong AHT (2019) Enhancing remote sensing image contrast based on combination of fuzzy logic and local approach. *J Inf Hiding Multimed Signal Process* 10(4):488–499
 46. Venter G (2010) Review of optimization techniques. American Cancer Society. <https://doi.org/10.1002/9780470686652.eae495>
 47. Wang X, Jin Y, Schmitt S, Olhofer M (2020) An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Inf Sci* 519:317–331
 48. Wu JMT, Lin JCW, Tamrakar A (2019) High-utility itemset mining with effective pruning strategies. *ACM Trans Knowl Discov Data* 13(6):1–22
 49. Wu TY, Lee Z, Obaidat MS, Kumari S, Kumar S, Chen CM (2020) An authenticated key exchange protocol for multi-server architecture in 5g networks. *IEEE Access* 8:28096–28108
 50. Xue X, Chen J, Pan JS (2018) Evolutionary algorithm based ontology matching technique. Science Press, USA
 51. Yang XS, Deb S (2009) Cuckoo Search via Lévy flights. In: 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), IEEE, pp 210–214. <https://doi.org/10.1109/NABIC.2009.5393690>
 52. Zhao L, Gai M, Jia Y (2018) Classification of multiple power quality disturbances based on PSO-SVM of hybrid kernel function. *J Inf Hiding Multimed Signal Process* 10(1):138–146
 53. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl-Based Syst* 163:283–304. <https://doi.org/10.1016/j.knosys.2018.08.030>
 54. Zhuang J, Luo H, Pan TS, Pan JS (2020) Improved flower pollination algorithm for the capacitated vehicle routing problem. *J Netw Intell* 5(3):141–156

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.