J Bionic Eng 18 (2021) 674–710
DOI: https://doi.org/10.1007/s42235-021-0050-y *Journal of Bionic Engineering*
http://www.springer.com/journal/42235

The Colony Predation Algorithm

Jiaze Tu[1], Huiling Chen[1*], Mingjing Wang[1], Amir H. Gandomi[2]

1. *College of Computer Science and Artificial Intelligence*, *Wenzhou University*, *Wenzhou* 325035, *China*
2. *Faculty of Engineering and Information Technology*, *University of Technology Sydney*, *Ultimo*, *NSW* 2007, *Australia*

Abstract

This paper proposes a new stochastic optimizer called the Colony Predation Algorithm (CPA) based on the corporate predation of animals in nature. CPA utilizes a mathematical mapping following the strategies used by animal hunting groups, such as dispersing prey, encircling prey, supporting the most likely successful hunter, and seeking another target. Moreover, the proposed CPA introduces new features of a unique mathematical model that uses a success rate to adjust the strategy and simulate hunting animals' selective abandonment behavior. This paper also presents a new way to deal with cross-border situations, whereby the optimal position value of a cross-border situation replaces the cross-border value to improve the algorithm's exploitation ability. The proposed CPA was compared with state-of-the-art metaheuristics on a comprehensive set of benchmark functions for performance verification and on five classical engineering design problems to evaluate the algorithm's efficacy in optimizing engineering problems. The results show that the proposed algorithm exhibits competitive, superior performance in different search landscapes over the other algorithms. Moreover, the source code of the CPA will be publicly available after publication.

Keywords: Colony Predation Algorithm, optimization, nature-inspired computing, meta-heuristic, engineering problems

1 Introduction

Optimization methods are not limited to single-objective methods, and every single objective idea can be extended for dealing with more classes of problems that have more than one or many objective functions. Common optimization approaches include fuzzy optimization[1], large scale problem solving[2], memetic and hybrid approaches[3], multi-objective optimization (an extension of the single-objective methods)[4], robust optimization, and many objectives[5]. Swarm-based stochastic methods involve any type of mathematical form and various inspirations. In recent years, metaheuristic algorithms[4] have attracted much attention and have been extensively used in numerous fields[6–17]. Such popularity is attributed to the ability of MAs to solve many possible complex feature spaces in practical problems in neural network-based control[18,19], formation control[20], deep learning models and feature understanding[21,22], adaptive control[23,24], machine learning-based implements[25,26], and artificial intelligence[27]. In addition to their characteristic continuity, discreteness, and constraints, MAs can also avoid local optimum, exhibit simplicity, and provide satisfactory solutions to complex problems without the need for gradient information[28,29].

Nowadays, metaheuristic algorithms have gained momentum in new engineering and technical problems[30]. In a previous work, we developed numerous MAs motivated by the behavior of biological and physical systems in nature. Metaheuristics can be divided into four categories[31]: Evolutionary Algorithms (EAs), Physics-based algorithms, Human-based algorithms, and Swarm Intelligence (SI) algorithms. Specifically, Holland proposed the Genetic Algorithm (GA), one of the most popular EAs[32], based on Darwin's theory of biological evolution. GA simulates the process of biological evolution, then, searches for the optimal solution in a solution space. The Differential Evolution (DE) algorithm[33] is another popular EA that simulates the cooperative relationship between individuals within a group and the swarm intelligence of competitive production to guide the direction of an optimization search. The stochastic components[34] contribute more variety to the searching patterns in DE. Other established EAs; include Genetic Programming (GP)[35], Evolution

***Corresponding author:** Huiling Chen
E-mail: chenhuiling.jlu@gmail.com

Strategy (ES)[36], and Evolutionary Programming (EP)[37]. Physics-based algorithms are inspired by the physical laws such as Simulated Annealing (SA)[38], which simulates the annealing process of metals in searching for the optimal solution of a problem. The Central Force Optimization (CFO) and Gravitational Search Algorithms (GSA)[39] are some other physics-based metaheuristics. Among the SI algorithms, Grey Wolf Optimizer (GWO)[40], Bat-inspired algorithm (BA)[41], Cuckoo Search (CS)[42], Artificial Bee Colony (ABC)[43], Slime Mould Algorithm (SMA)[44], Harris Hawks Optimization (HHO)[45], Hunger Games Search (HGS)[46], Krill Herd (KH)[47], Moth Search Algorithm (MSA)[48], Monarch Butterfly Optimization (MBO)[49], Moth Flame Optimization (MFO)[50], Marine Predators Algorithm (MPA)[51], and Whale Optimization Algorithm (WOA)[52] are widely used. Some human-based optimizers include Tabu Search (TS)[53] and Teaching Learning-based Optimization (TLBO)[54,55]. While metaheuristic algorithms have their own advantages and disadvantages compared to alternative solvers, they all provide the benefits of simplicity and relatively fast running time.

MAs are divided into four categories that share two principal characteristics: exploration and exploitation. In brief, the randomness of the search is essential to explore the search space as much as possible to ensure that the algorithm has enough exploration capability; thus exploration shows the ability and richness of randomness. Then, the exploitation stage is based on a promising area achieved by the exploration phase, focusing on the local search aptitude. Finding an outstanding balance between these two stages is one of the most challenging metaheuristics problems, which directly relates to the algorithm's performance. Therefore, first-class algorithms, such as DE and ABC are the optimal approaches to maintain the right balance between exploration and exploitation.

Even though numerous excellent algorithms have been proposed, the No Free Lunch (NFL) theorem[56] in search indicates shows that none of these methods is a universal best technique for solving any existing or future problems. In other words, each algorithm can only solve one or a class of optimization problems. Therefore, this work developed an effective metaheuristic algorithm, the Colony Predation Algorithm (CPA) based on the coexistence of animals. Specifically, CPA mimics the supportive behavior of social animals and predation strategy of hunting animals. In order to improve its superiority, we applied CPA to engineering problems and achieved good results.

The paper is structured as follows. Section **2** presents the background and inspiration for CPA, including its formula, pseudocode, and time complexity. Section **3** discusses the results of the CPA in solving different benchmark problems and the selection of CPA coefficients. Section **4** describes the experiments performed on engineering design problems using the developed algorithm. Section **5** summarizes and concludes the paper.

2 Colony predation algorithm

This section introduces the inspiration for the CPA and establishes its mathematical model.

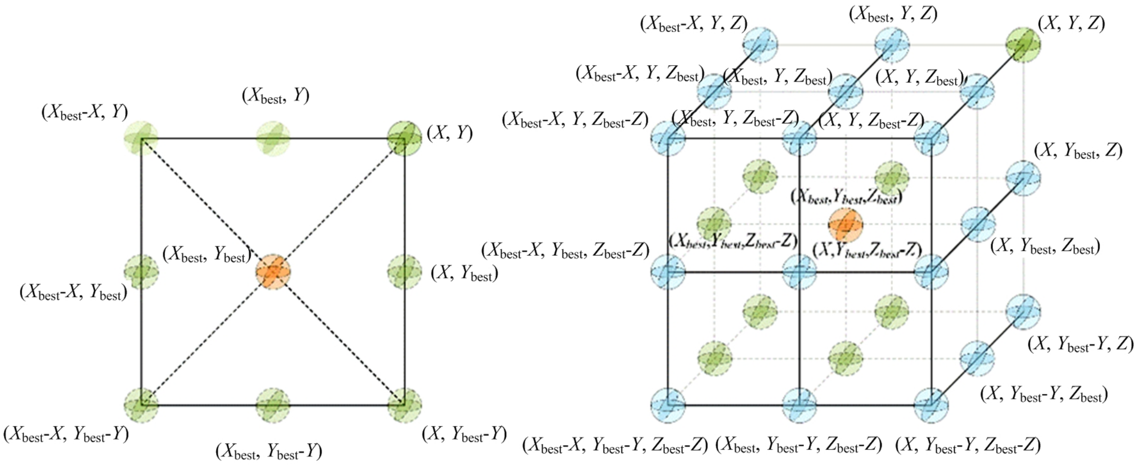
2.1 Inspiration

To develop our algorithm, we considered the behavior of social animals. Specifically, we focused on colony predation, which is performed by animals to avoid enemies, and increase the probability of hunting success[57]. This strategy is commonly portrayed by wolves, hyenas, lions, piranha, and other animals that prey in groups rather than individually[58].

Colony predation can often help predators acquire more prey, thus increasing individual survival rates. Animals living in groups perform colony predation because their survival is closely related to the group, and the behavior of calling partners also makes their search more efficient. For example, hyenas and wolves communicate and cooperate *via* colony predation to catch more prey than individuals could[59].

Dividing and surrounding prey are the two most commonly used methods that improve the probability of hunting success. Selective abandonment is another strategy that animals adopt when they encounter situations, such as when consumption exceeds their harvest[60]. In this behavior, they will abandon their target and choose another which makes their predation more efficient[61].

Herein, we propose a rule for the survival of the



$(X_{best}-X, Y)$
(X_{best}, Y)
(X, Y)
(X_{best}, Y_{best})
$(X_{best}-X, Y_{best})$
(X, Y_{best})
$(X_{best}-X, Y_{best}-Y)$
$(X_{best}, Y_{best}-Y)$
$(X, Y_{best}-Y)$
$(X_{best}-X, Y, Z)$
(X_{best}, Y, Z)
(X, Y, Z)
$(X_{best}-X, Y, Z_{best})$
(X_{best}, Y, Z_{best})
(X, Y, Z_{best})
$(X_{best}-X, Y, Z_{best}-Z)$
$(X_{best}, Y, Z_{best}-Z)$
$(X, Y, Z_{best}-Z)$
(X, Y_{best}, Z)
$(X_{best}, Y_{best}, Z_{best})$
$(X_{best}-X, Y_{best}, Z_{best}-Z)$
$(X_{best}, Y_{best}, Z_{best}-Z)$
$(X, Y_{best}, Z_{best}-Z)$
(X, Y_{best}, Z_{best})
$(X, Y_{best}-Y, Z)$
$(X, Y_{best}-Y, Z_{best})$
$(X_{best}-X, Y_{best}-Y, Z_{best}-Z)$
$(X_{best}, Y_{best}-Y, Z_{best}-Z)$
$(X, Y_{best}-Y, Z_{best}-Z)$

(a) (b)

Fig. 1 Possible locations in 2 and 3-dimensions.

fittest, where the loser is replaced with the winner. More specifically, the leaders of lion and wolf packs are those who win a fight with the previous leader.

2.2 Mathematical model

The mathematical simulation of the position of the algorithm is given here.

Fig. 1 displays the search process of groups and individuals in two and three dimensions, where a predator at position (X, Y) can update its position according to the target's position (X_{best}, Y_{best}).

Fig. 2 shows how the search agent updates its position based on the predator leader and other predators in the 2D search space. It can be observed that the final position will be a random position within the circle defined by the positions of the predator leader and other predators in the search space. The gray circle represents the final direction of the updated position.

2.2.1 Communication and collaboration

Animals who hunt in groups have an increased success rate of predation through communication and cooperation. The following formulas represent individual cooperative communication and food searching behavior:

$$\vec{X}_j^i(t+1)=\vec{X}_j^{i_1}(t)+(1-r)\cdot((\vec{X}_1(t)+\vec{X}_2(t))/2), \qquad (1)$$

where r is in the range of [0, 1], $\vec{X}_j^{i_1}(t)$ is the individual looking for food; $\vec{X}_1$ and $\vec{X}_2$ are the two closest positions to prey in the j-th dimension; $j\in 1,2,...,dim$; and $\vec{X}_j^i(t+1)$ is the latest updated position of the individual.

2.2.2 Disperse food

The first strategy in colony predication drives the prey in different directions, separating the prey from its group. The predation strategy displayed by individuals in search is simulated mathematically as follows:

$$\vec{X}(t+1)=\vec{X}_{best}-S\cdot(\boldsymbol{r}_1\cdot(ub-lb)+lb), \qquad (2)$$

where $\vec{X}(t+1)$ is the position of a population; $\vec{X}_{best}$ is the position of food; S represents the strength of prey, where its absolute value decreases from a to 0 with the number of evaluations, r_1 represents $[R_1;R_2;R_3;...R_j]$; and $[R_1;R_2;R_3;...R_j]$ is the random number of [0,1]; $j=dim$ is the dimension of the population; and, ub, and lb are the upper and lower bounds, respectively.

The formula of S is as follows:

$$S_0=a-t\cdot(\frac{a}{N}), \qquad (3)$$

$$S=2\cdot S_0\cdot r_2-S_0, \qquad (4)$$

where N represents the number of individuals; S_0 decreases from a to 0 with the number of evaluations; t represents the current number of evaluations; and, r_2 is a random number of [0,1].

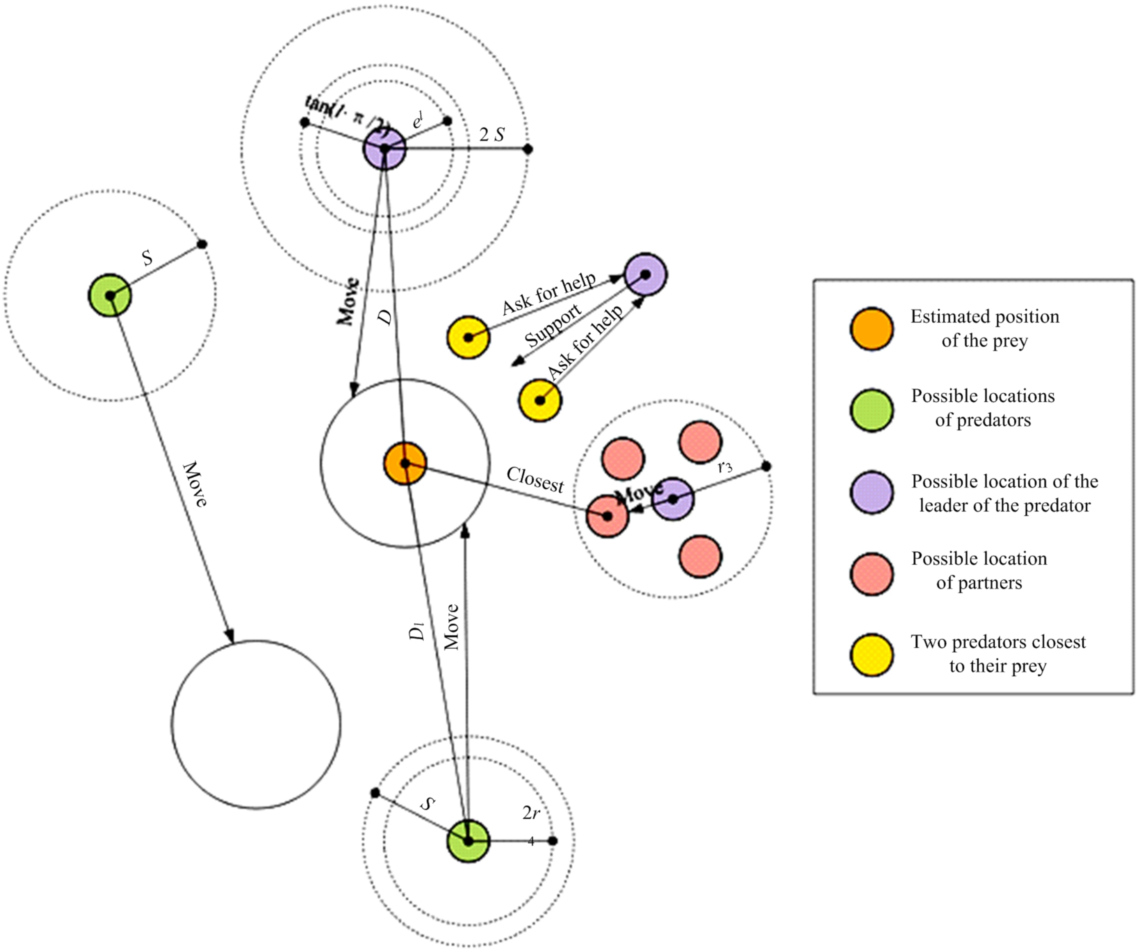

tan(l · π /2)
e^l
2 S
S
Move
D
Ask for help
Support
Ask for help
Closest
r3
Move
D1
Move
S
2r
Estimated position of the prey
Possible locations of predators
Possible location of the leader of the predator
Possible location of partners
Two predators closest to their prey

Fig. 2 Position updating in CPA.

The formula of a is as follows:

$$a = e^{w-2w\cdot(1-\frac{t}{MaxFEs})}, \tag{5}$$

where $w = 9$.

2.2.3 Encircle food

The hunting group will use the second strategy to surround and approach the prey. This stage can be represented by mathematical simulation as:

$$\vec{X}(t+1) = \vec{X}_{best} - 2S\cdot\boldsymbol{D}\cdot e^{l}\cdot\tan(\frac{\pi}{4}l), \tag{6}$$

where $\boldsymbol{D}$ is the distance between the current individual and prey, and $\boldsymbol{D}$ is different between different individuals; l is a random number of [0,1]; and, $\tan(\frac{\pi}{4}l)$ is the encirclement curve of the hunter.

The formula of $\boldsymbol{D}$ is as follows:

$$\boldsymbol{D} = \left|\vec{X}_{\text{best}} - \vec{X}(t)\right|, \tag{7}$$

where $\vec{X}(t)$ represents the current hunter population.

The execution probabilities of these two predatory strategies are expressed as:

$$\vec{X}(t+1) = \begin{cases} \vec{X}_{\text{best}} - S\cdot(r_1\cdot(ub-lb)+lb) & r_7 \geq 0.5 \\ \vec{X}_{\text{best}} - 2S\cdot\boldsymbol{D}\cdot e^{l}\cdot\tan(\frac{\pi}{4}l) & r_7 < 0.5 \end{cases} \tag{8}$$

2.2.4 Supporting closest individual

Considering that the group may encounter difficulties in hunting prey, the nearest individual calls for peer support, which can be expressed as follows:

$$\vec{X}(t+1) = \vec{P}_{\text{nearest}}, \tag{9}$$

where $\vec{P}_{\text{nearest}}$ is the location of the nearest predator in the support group; and $\vec{P}$ is the predator closest to the prey.

The formula of $\vec{P}$ is as follows:

$$\vec{P} = r_3 \cdot \vec{X}_j^i, \tag{10}$$

where $j \in 1,2,...,dim$; r_3 represents $[R_1;R_2;R_3;...R_j]$ and $[R_1;R_2;R_3;...R_j]$ is the random number of [0,1].

2.2.5 Searching for the food

The remaining individuals will find an alternative food source if no prey is found nearby or food too far away from the prey. This behavior can be expressed as follows:

$$\boldsymbol{D}_1 = \left|2r_4 \cdot \vec{X}_{\text{rand}} - \vec{X}(t)\right|, \tag{11}$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - S \cdot \boldsymbol{D}_1, \tag{12}$$

where $\boldsymbol{D}_1$ denotes the distance of random group movement, r_4 is a random number of [0,1], and $\vec{X}_{\text{rand}}$ is a new individual position formed randomly by individuals.

The formula of $\vec{X}_{\text{rand}}$ is as follows:

$$\vec{X}_{\text{rand}} = r_5 \cdot ((ub - lb) + lb), \tag{13}$$

where ub and lb are the upper and lower bounds of functions, respectively, dim is the dimension of population; r_5 represents $[R_1;R_2;R_3;...R_j]$; and $[R_1;R_2;R_3;...R_j]$ is the random number of [0,1].

The probability of implementing supporting the closest individual and searching for the food is determined by the r_6 between the group and prey, which can be expressed as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{P}_{\text{nearest}} & |r_6| \le 1 \\ \vec{X}_{\text{rand}} - S \cdot \boldsymbol{D}_1 & |r_6| > 1 \end{cases}, \tag{14}$$

where r_6 is a random number of [−2, 2].

Eq. (10) and Eq. (12) describe the randomness so the searched solution can traverse the solution space. Specifically, Eq. (10) utilizes the current solution as a reference and randomly generates multiple random solutions, so that the current solution is replaced by the generated optimal solution to achieve the purpose of avoiding the local optimum. The current solution is then updated by a completely random solution and replaced by the current solution in Eq. (12), which is equivalent to searching randomly for fresh prey when the current prey cannot be captured.

When the predator's position exceeds the upper or lower limit, we introduce the rule of survival of the fittest in nature and replace the position beyond the limit with the current optimal position. The specific formula is as follows:

$$\vec{X}_{\text{ofb}}(j) = \vec{X}_{\text{best}}(j), \tag{15}$$

where $\vec{X}_{\text{ofb}}(j)$ is the position beyond the boundary and $\vec{X}_{\text{best}}(j)$ is the optimal position, $j \in 1,2,...,dim$.

The CPA proposed in this paper imitates the process of colony predation. We simplify the algorithm as much as possible to maximize its scalability. Algorithm 1 shows the pseudo-code of the CPA.

Fig. 3 displays the flowchart of CPA, where $MaxFEs$ represents the maximum number of evaluations, $value$ indicates the fitness of each evaluation, N denotes the population number, and dim is the problem's dimension.

As a unique optimizer with stable performance, CPA has exhibited high potential to solve optimization cases, which is attributed to the following reasons:

(1) The idea of selective abandonment adopted in the algorithm breaks the boundary between exploration and exploitation and increases the exploratory tendency even in the middle and later stages of exploitation, which further helps to prevent from dropping into the Local Optimum (LO).

(2) The value of the current optimal solution is used rather than the value of transboundary individuals, which avoids the problem of excessive spatial dispersion.

(3) The introduction of adaptive weight a ensures that the algorithm can quickly transit from exploration to exploitation in the early stage, allowing more time to execute the exploitation, while S guarantees the perturbation of the algorithm so it will not fall into the LO prematurely.

(4) Innovative use of the communication and cooperation mechanism can increase the diversity in the early stage and strengthen exploration of the local solution space in the later stage.

Algorithm 1 Pseudo-code of CPA

```
Initialize the parameters $N$, $MaxFEs$, $dim$, $value$
Initialize the positions of Individuals $\vec{X}_i (i = 1, 2, ..., N)$
While( $t \le MaxFEs$ )
    For $i = 1: N$ do
        When an individual's position exceeds the $ub$ and $lb$, using Eq. (15) to replace the position beyond the boundaries
        Calculate the fitness of all Individuals
        Update the $\vec{X}_{\text{best}}$
    End For
    Update the $S$ by Eq. (3) and Eq. (4)
    Update the $a$ by Eq. (5)
    Update the $r_1$
    For $j = 1:dim$ do
        Update the $\vec{X}_1$, $\vec{X}_2$, $i_1$, $r$
        If $r < \frac{t}{MaxFEs}$ do
        $\vec{X}_i = \vec{X}_{best}$
        End If
        Calculate the $\vec{X}_j^i$ by Eq. (1)
    End For
    For $i = 1: N$ do
        Update $S$, $l$
        If $|S| < \frac{2}{3}a$ do
            Calculate the $\vec{X}_i$ by Eq. (8)
        Else
            Calculate the $\vec{X}_i$ by Eq. (14)
        Endif
    End For
    $t = t + 1$
End While
Return $\vec{X}_{\text{best}}$
```

Fig. 4 further demonstrates the authenticity of the preceding ideas. First, we chose test function, F1, to prove point 2 and F12 to prove the influence of points 1, 3, and 4 on CPA's exploration ability. In the figure, CPA is in red, CPA with linear parameter a is in blue, CPA with the communication and cooperation mechanism removed is in green, and CPA with the abandonment mechanism removed is pink. Comparing (d), (c), and (d) proves that the current optimal value can effectively reduce the distribution of space midpoint and improve the performance. It can be seen from (f), (g), and (h), it can be seen that the CPA variants fall into the LO earlier than CPA on its own, thus proving that points 1, 3, and 4 are correct.

2.2.6 Computational complexity analysis

In the proposed CPA initialization, fitness evaluation, communication and collaboration, parameter updating, and location updating are performed. In the respective functions, N is the number of individuals in the population, D is the dimension of the problem, and $MaxFEs$ indicates the maximum number of evaluations. $O(N)$ refers to the computational complexity of initialization, fitness evaluation, parameter updating, and communication and collaboration, while the computational complexity of location updating is $O(N \times (1 + \frac{D + N^2}{4}))$. From this, we can obtain the complexity of the whole algorithm: $O(N \times (1 + MaxFEs \times N \times (1 + \frac{D + N^2}{4})))$.

3 Experiments and results

This section compares the proposed CPA with a number of conventional and recent optimizers in the

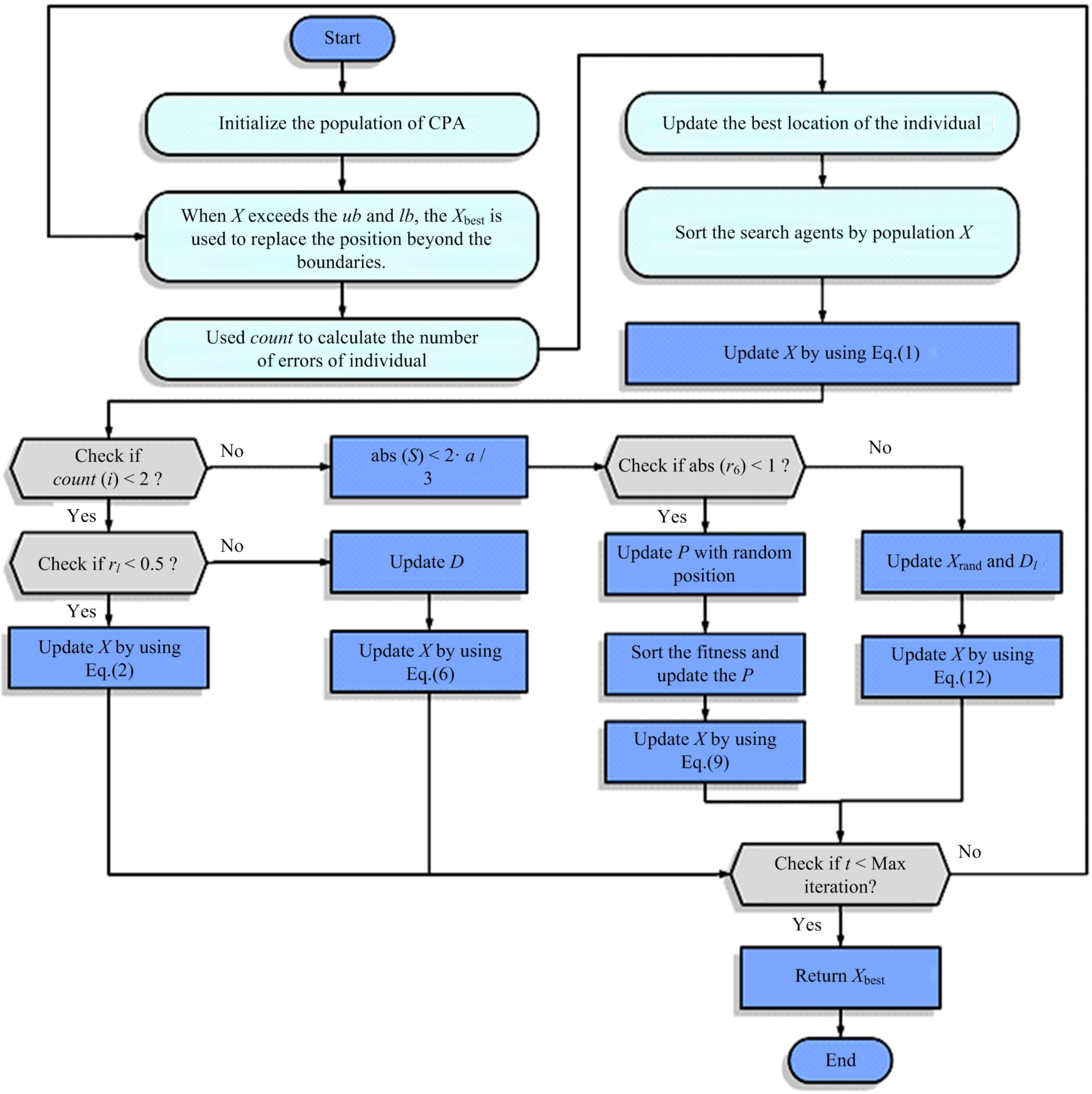

Start
Initialize the population of CPA
When X exceeds the ub and lb, the X_{best} is used to replace the position beyond the boundaries.
Used $count$ to calculate the number of errors of individual
Update the best location of the individual
Sort the search agents by population X
Update X by using Eq.(1)
Check if $count$ (i) < 2 ?
No
abs (S) < 2· a / 3
Check if abs (r_6) < 1 ?
No
Yes
Yes
Check if r_l < 0.5 ?
No
Update D
Update P with random position
Update X_{rand} and D_l
Yes
Update X by using Eq.(2)
Update X by using Eq.(6)
Sort the fitness and update the P
Update X by using Eq.(12)
Update X by using Eq.(9)
Check if t < Max iteration?
No
Yes
Return X_{best}
End

Fig. 3 Flowchart of CPA.

field. All experiments were conducted on Windows Server 2008 R2 operating system with Intel (R) Xeon (R) CPU E5-2650 v4 (2.20 GHz) and 128 GB of RAM. We coded all algorithms for comparison using MATLAB R2014b.

3.1 Benchmark function validation

The algorithm was tested on 53 functions, among which F1~F23 are benchmark functions (Table A1) and F24~F53 are CEC2014 functions (Table A2). These functions cover both the monomodality and multimodality of problems. In the respective functions, Dim represents the dimension of the function, Range refers to the definition domain of the function, and $f_{\min}$ reveals the optimal solution of the function.

All experiments were conducted under the same conditions to ensure the fairness of the experimental results[62–64]. Under the evaluation framework, the population size was set to 30, and the number of evaluations and dimensions were set to 1000 and 30, respectively. This ensures that there is no partiality of the system, as per literature[26,65–72]. At the same time, we tested each algorithm 30 times to exclude the influence of random factors. We applied the Friedman test and the Wilcoxon-signed rank test to compare the performances of the algorithms. The Friedman test is a non-parametric

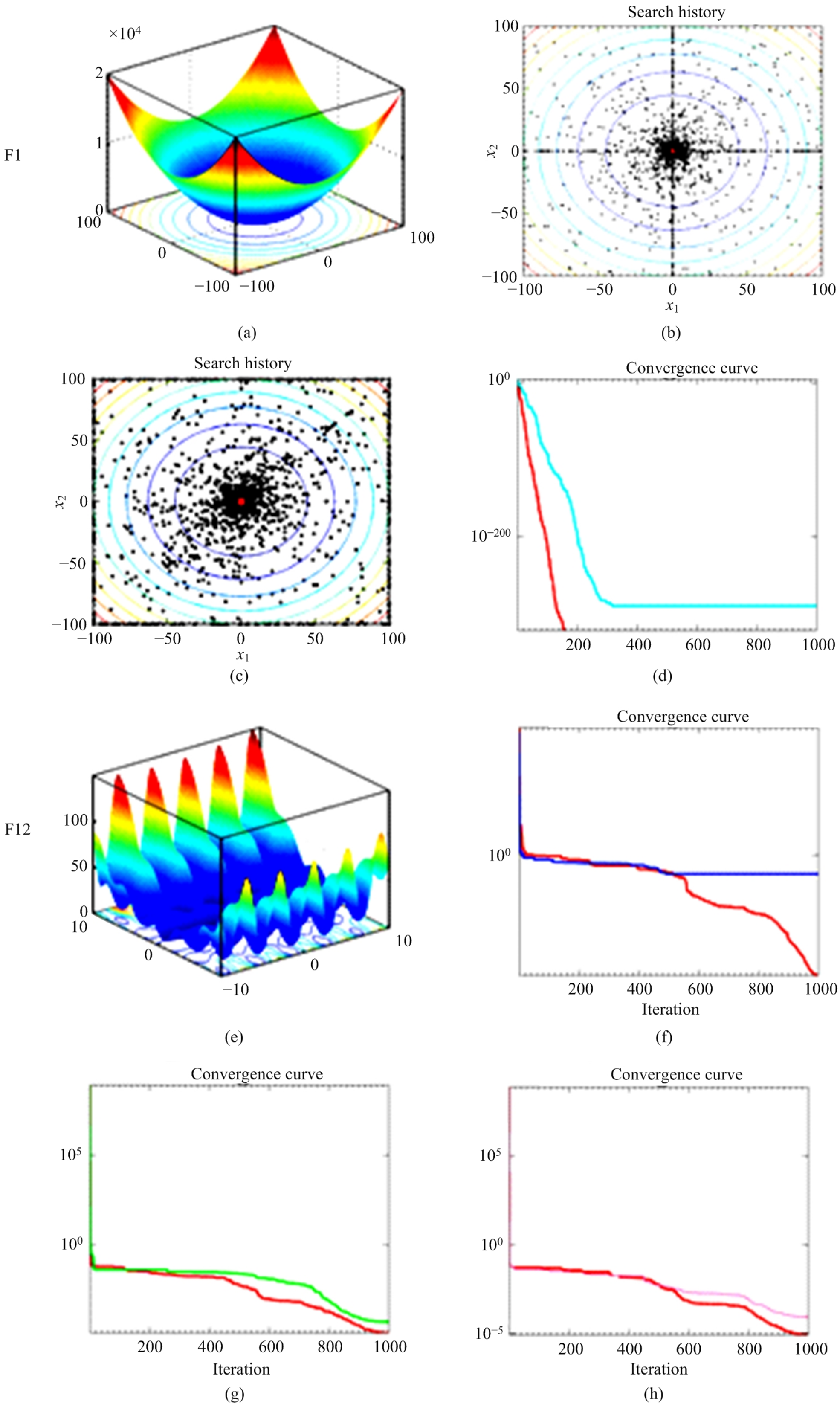


Fig. 4 Comparison of the mechanisms of each part of CPA: (a) Three-dimensional image of function; (b) lattice diagram of CPA; (c) CPA that uses the boundary value to process out of boundary; (d) comparison diagram of the convergence curve, in which red is the CPA.

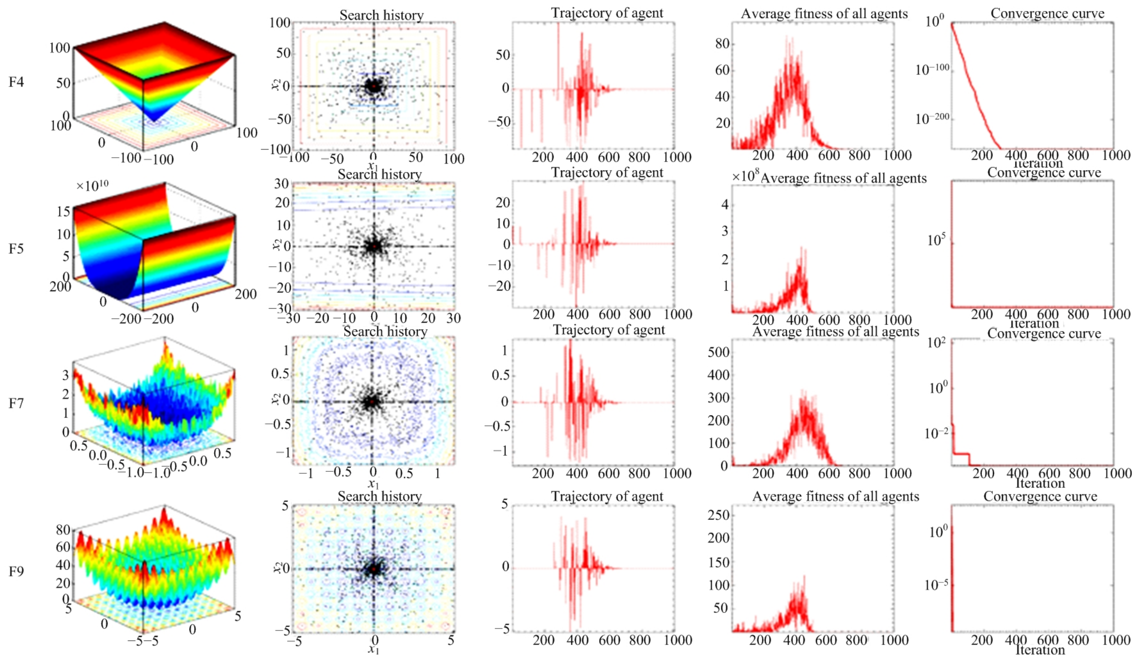


Fig. 5 Qualitative analysis of CPA.

statistical program that allows us to do further analysis of the algorithm's average performance ranking. The Wilcoxon test is often used for statistical testing, in which a p-value less than 0.05 indicates that the performance of CPA performance is better than its competitors.

3.2 Qualitative analysis

Fig. 5 presents the results of a qualitative analysis of CPA on benchmark functions. F4, F5, F7 and F9 (from Table A1) to analyze its exploration and exploitation ability. We selected 2 single-mode functions and 2 multi-mode functions as the evaluation criteria. The results consist of four major components: 1) The search history shows the location and distribution of individuals in each evaluation; 2) the trajectory of the first individual reveals the individual's motion law throughout the evaluation process; 3) the average fitness of all individuals monitors how the average fitness of the entire population changes during the optimization process; and 4) the convergence behavior reveals the changing trend of optimal fitness.

We can see that an individual explored most of the solution space according to its historical position, revealing that the algorithm has strong search ability and can avoid falling into local optimal solutions on complex multimode functions. At the same time, we can also observe that most of the search locations are around the optimal solution, which suggests that the algorithm can be accurately developed in the target area. Moreover, the convergence speed is quick on all functions except F4, yet it can still reach the medium term's optimal value.

The trajectory diagram shows that individuals have sharp fluctuations in the initial stage and the medium term of search. The fluctuation coverage exceeds 50% of the solution space, proving that the CPA has a powerful search ability. The algorithm can find the optimal solution quickly from the function image and, thus, performs very well on simple single-mode or complex multimode functions.

The algorithm tends to converge quickly in the early stages of evaluation by monitoring the overall average fitness. CPA obtained the best average fitness of all functions, except F15 and F23, in a concise time with repetition. Although the average fitness fluctuated at times, its gradual decrease reflects the algorithm's super search and pioneering ability. The convergence curve further reveals the algorithm's fast convergence speed.

3.2.1 Comparison with other algorithms

CPA was further compared with 11 traditional MAs, including SCA[73], SSA[74], GWO[40], MFO[50], WOA[52], ABC, GSA, PSO[75], DE, FA[76], and BA. Table A3 lists the parameters of these algorithms, which were taken from the classical algorithm without any modification.

The data in Table 1 provides the results of CPA compared with other traditional MAs based on average results and ranking of the Friedman test in the last column. In the table, "+" indicates that the CPA's performance is better than that of the corresponding algorithm; "–" means that CPA performance is worse than the corresponding algorithm; and "=" CPA and the corresponding algorithm. The average (AVG) and standard deviation (STD) values were obtained from the Friedman test and correspond to the algorithm's average ranking result. We can intuitively find from Table 1showsthat CPA ranks first and that it was challenging for the competitive algorithms to defeat CPA on most of the 53 functions. In addition, the Friedman rank of CPA is only 3.20, which is much smaller than the other algorithms. Compared with the second DE, CPA's average is about 0.9 lower than DE. CPA also ranks in the top three on F6, F13, F21, F22, F23, F25, F26, F27, F29, F32, F34, and F38. Moreover, because the STD of CPA reached 0 on F1, F2, F3, F4, F9, F17, F47, and F48, we can conclude that the CPA has excellent performance.

Table A4 shows the results of the Wilcoxon test most of the *p*-values are less than 0.05, accounting for 88% of all data. Even in SCA, only two p-values are higher than 0.05. CPA is far superior to the DE and ABC on most functions, although the number of *p*-values higher than 0.05 accounts for 13% and 19% of the total, respectively. This further shows that the CPA has a strong statistical significance.

Fig. 6 reveals that the convergence speed of the CPA is very fast, which provides an index to judge the algorithm's performance. CPA found the optimal value in the early stage of F1 and F4, so there is no CPA curve in the image. For F1, F4, F7, F15, F46, F50, and F53, CPA also converged the fastest among all algorithms, while some other algorithms converged quite slow, and fell into local optima. Functions F46, F50, and F53 demonstrate that CPA has high accuracy in solving problems and can quickly find the global optimum at the beginning of the evaluation. While the convergence speed of some other algorithms was also very fast, their accuracy in finding the solution was not as high as CPA's, and the solution found by CPA is better. Even though its CPA convergence speed was slower on F24, CPA still found the global optimum before other algorithms; some of which even fell into the local optimum at the beginning of the evaluation. It can be concluded from the performance on F15 and F35 functions that CPA has a strong ability for global exploration, can find the target area in the early stage and fully balances the exploration and exploitation, further confirming its superior performances.

3.2.2 Comparison with advanced MAs

To further illustrate its superior CPA was also compared with 9 well-known improved algorithms: Modified Sine Cosine Algorithm (m_SCA)[77], Opposition-Based Sine Cosine Algorithm (OBSCA)[78], Modified Whale Optimization Algorithm (MWOA)[79], Improved Whale Optimization Algorithm (IWOA)[80], Bat Algorithm based on Collaborative and Dynamic Learning of Opposite Population (CDLOBA)[81], Chaotic Bat Algorithm (CBA)[82], Chaotic and Gaussian Particle Swarm Optimization (CGPSO)[83], Particle Swarm Optimization with an Aging Leader and Challengers (ALCPSO)[84], and Differential Evolution Algorithm based on Chaotic Local Search (DECLS)[85].

The data in Table 2 reveal that the CPA has an excellent performance in both single-mode and multimode functions, especially in the fixed dimension multimode function. The average of CPA results is 2.88, which is much smaller than the average of other algorithms. For instance, this average is one-fifth the average of MWOA, m_SCA, OBSCA, IWOA, CDLOBA, CBA, and CGPSO algorithms, the strongest algorithms defeated CPA on only 11 out of the 53 functions, and MWOA did not beat CPA on any function. This shows that the CPA has a strong optimization ability.

Table A5 lists the *p*-values of CPA and the advanced algorithm on all test functions and shows that all values of MWOA are less than 0.05. We can also see that the difference between the values is more than 0.05, although the number of ALCPSO's values greater than 0.05 approaches 14. These test results further confirm that the CPA is superior to other competitive algorithms.

F1, F4, F10, and F20 belong to 23 benchmark functions, while F26, F28, F35, F39, F43, F50, F52, and

Table 1 Comparison results of CPA with traditional algorithms on the 53 functions

Algorithm	F1		F2		F3	
	AVG	STD	AVG	STD	AVG	STD
CPA	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
PSO	9.648E+01	1.103E+01	4.569E+01	3.989E+00	1.703E+02	2.269E+01
DE	1.07E−241	0.000E+00	1.95E−142	2.21E-142	2.655E+02	1.352E+02
ABC	4.881E−16	4.751E−17	1.209E−15	1.135E−16	8.885E+02	3.661E+02
SCA	4.993E−85	2.732E−84	2.216E−86	1.182E−85	8.397E−02	3.090E−01
SSA	3.040E-09	7.088E−10	1.025E−01	1.784E−01	3.447E−08	1.075E−08
BA	6.230E−01	3.947E−01	2.626E+00	9.802E−01	2.328E−01	2.311E−01
FA	1.117E+04	1.084E+03	4.761E+01	3.679E+00	1.839E+04	2.322E+03
MFO	2.667E+03	5.208E+03	3.400E+01	2.207E+01	1.529E+04	1.135E+04
WOA	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.185E+00	5.625E+00
GWO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	5.74E−272	**0.000E+00**
GSA	1.441E+01	1.639E+00	1.520E+01	8.396E−01	2.801E+02	5.079E+01
Algorithm	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CPA	**0.000E+00**	**0.000E+00**	**2.143E+01**	**1.578E−01**	2.115E−26	2.834E−26
PSO	3.638E+00	2.122E−01	7.769E+04	1.557E+04	9.153E+01	9.514E+00
DE	1.535E−21	7.511E−21	3.085E+01	1.921E+01	**0.000E+00**	**0.000E+00**
ABC	1.434E+00	4.321E−01	2.382E-01	5.617E-01	4.795E−16	7.891E−17
SCA	2.310E−04	1.198E−03	2.731E+01	6.822E−01	3.485E+00	2.589E−01
SSA	1.578E−01	3.946E−01	3.753E+01	3.316E+01	3.006E−09	7.252E−10
BA	4.530E+00	6.079E+00	2.840E+02	4.338E+02	6.433E−01	3.621E−01
FA	3.930E+01	2.088E+00	7.156E+06	1.402E+06	1.090E+04	1.164E+03
MFO	6.410E+01	9.421E+00	2.689E+06	1.459E+07	1.010E+03	3.082E+03
WOA	3.516E+00	1.113E+01	2.372E+01	2.050E−01	1.116E−06	4.410E−07
GWO	3.13E−230	**0.000E+00**	2.600E+01	9.075E−01	4.999E−01	3.280E−01
GSA	1.701E+00	1.279E−01	8.600E+03	1.545E+03	1.457E+01	1.887E+00
Algorithm	F7		F8		F9	
	AVG	STD	AVG	STD	AVG	STD
CPA	**2.706E−05**	7.163E−05	**−1.257E+04**	5.145E−12	**0.000E+00**	**0.000E+00**
PSO	1.130E+02	2.615E+01	−7.026E+03	9.487E+02	3.380E+02	1.904E+01
DE	1.633E−03	3.467E−04	−1.245E+04	1.190E+02	1.327E−01	3.440E−01
ABC	6.528E−02	1.524E−02	**−1.257E+04**	**2.925E-12**	**0.000E+00**	**0.000E+00**
SCA	1.621E−03	1.522E−03	−4.481E+03	2.686E+02	**0.000E+00**	**0.000E+00**
SSA	5.192E−03	1.976E−03	−7.686E+03	6.040E+02	6.978E+01	1.821E+01
BA	1.400E+01	1.200E+01	−7.093E+03	6.521E+02	2.419E+02	2.211E+01
FA	3.794E+00	6.533E−01	−4.212E+03	2.135E+02	2.239E+02	1.247E+01
MFO	1.420E+00	2.510E+00	−8.692E+03	9.020E+02	1.543E+02	4.011E+01
WOA	1.059E−04	1.075E−04	−1.228E+04	7.883E+02	**0.000E+00**	**0.000E+00**
GWO	3.734E−05	**2.772E−05**	−6.223E+03	6.377E+02	**0.000E+00**	**0.000E+00**
GSA	2.970E+01	4.650E+00	−2.698E+03	5.512E+02	1.969E+02	8.544E+00
Algorithm	F10		F11		F12	
	AVG	STD	AVG	STD	AVG	STD
CPA	**8.882E−16**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**4.731E−29**	1.215E−28
PSO	7.761E+00	2.379E−01	1.007E+00	1.564E−02	3.243E+00	3.976E−01

DE	7.520E−15	1.228E−15	0.000E+00	0.000E+00	1.571E−32	**5.567E−48**
ABC	3.784E−14	3.178E−15	1.110E−16	1.010E−16	4.685E−16	5.305E−17
SCA	1.253E+01	8.896E+00	6.883E−04	3.770E−03	3.142E−01	5.660E−02
SSA	1.708E+00	7.414E−01	1.370E−02	1.151E−02	9.638E−01	1.748E+00
BA	1.766E+00	7.075E−01	1.381E−02	1.907E−02	7.716E+00	3.099E+00
FA	1.583E+01	2.980E−01	1.006E+02	1.008E+01	1.854E+06	8.197E+05
MFO	1.216E+01	9.045E+00	3.013E+01	6.424E+01	1.837E−01	3.564E−01
WOA	2.783E−15	2.030E−15	**0.000E+00**	**0.000E+00**	1.969E−07	8.899E−08
GWO	7.401E−15	1.347E−15	**0.000E+00**	**0.000E+00**	2.672E−02	1.114E−02
GSA	4.122E+00	1.492E−01	5.166E−01	5.058E−02	1.315E+00	2.810E−01
Algorithm	F13		F14		F15	
	AVG	STD	AVG	STD	AVG	STD
CPA	5.489E-28	5.171E-28	**9.980E-01**	**1.304E-16**	**3.075E-04**	**7.282E-17**
PSO	1.503E+01	1.510E+00	2.383E+00	1.669E+00	9.506E-04	5.628E-05
DE	**1.350E-32**	**5.567E-48**	1.031E+00	1.815E-01	9.801E-04	3.661E-03
ABC	4.594E-16	6.662E-17	**9.980E-01**	0.000E+00	4.034E-04	7.263E-05
SCA	1.992E+00	1.686E-01	**9.980E-01**	5.240E-08	5.934E-04	4.165E-04
SSA	4.761E-03	5.538E-03	**9.980E-01**	2.020E-16	5.191E-04	3.684E-04
BA	1.373E-01	9.713E-02	2.614E+00	1.687E+00	3.487E-03	6.736E-03
FA	1.456E+07	4.711E+06	**9.980E-01**	6.804E-06	9.327E-04	1.455E-04
MFO	1.367E+07	7.487E+07	2.607E+00	2.493E+00	9.576E-04	5.088E-04
WOA	3.696E-04	2.006E-03	**9.980E-01**	6.331E-15	4.921E-04	3.719E-04
GWO	4.094E-01	1.845E-01	3.546E+00	3.765E+00	4.410E-03	8.118E-03
GSA	7.866E+00	1.265E+00	**9.980E-01**	1.400E-05	1.048E-03	2.185E-04
Algorithm	F16		F17		F18	
	AVG	STD	AVG	STD	AVG	STD
CPA	**−1.032E+00**	5.831E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	**1.072E−15**
PSO	**−1.032E+00**	8.495E−05	**3.979E−01**	3.811E−05	3.006E+00	5.354E−03
DE	**−1.032E+00**	6.775E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	1.951E−15
ABC	**−1.032E+00**	6.775E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	6.303E−07
SCA	**−1.032E+00**	1.771E−06	**3.979E−01**	4.054E−05	**3.000E+00**	1.129E−07
SSA	**−1.032E+00**	**4.225E−16**	**3.979E−01**	**0.000E+00**	**3.000E+00**	1.020E−14
BA	**−1.032E+00**	4.922E−05	**3.979E−01**	3.706E−05	3.005E+00	5.127E−03
FA	**−1.032E+00**	8.254E−05	**3.979E−01**	3.534E−05	3.001E+00	7.757E−04
MFO	**−1.032E+00**	6.775E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	1.424E−15
WOA	**−1.032E+00**	1.086E−15	**3.979E−01**	3.636E−11	**3.000E+00**	3.758E−08
GWO	**−1.032E+00**	1.368E−11	**3.979E−01**	5.214E−10	**3.000E+00**	4.630E−08
GSA	−1.031E+00	1.454E−04	3.981E−01	1.345E−04	3.005E+00	5.753E−03
Algorithm	F19		F20		F21	
	AVG	STD	AVG	STD	AVG	STD
CPA	**−3.863E+00**	2.518E−15	**−3.322E+00**	**1.477E−15**	−8.964E+00	2.193E+00
PSO	−3.857E+00	3.329E−03	−2.980E+00	1.751E−01	−7.350E+00	1.315E+00
DE	**−3.863E+00**	2.710E−15	−3.317E+00	1.972E−02	−9.648E+00	1.542E+00
ABC	**−3.863E+00**	**2.372E−15**	**−3.322E+00**	1.355E−15	−1.015E+01	**5.761E−15**
SCA	−3.855E+00	1.983E−03	−2.867E+00	4.086E−01	−2.571E+00	2.466E+00
SSA	−3.863E+00	1.820E−15	−3.235E+00	5.348E−02	**−9.816E+00**	1.282E+00
BA	−3.855E+00	2.919E−03	−3.056E+00	4.663E−02	−8.865E+00	1.598E+00
FA	−3.862E+00	2.164E−04	−3.258E+00	4.680E−02	−8.419E+00	6.918E−01

MFO	**−3.863E+00**	2.710E−15	−3.226E+00	5.648E−02	−5.307E+00	3.343E+00
WOA	−3.862E+00	1.999E−03	−3.220E+00	1.768Ev01	−1.015E+01	1.825E−07
GWO	**−3.863E+00**	1.439E−03	−3.253E+00	7.313E−02	**−9.816E+00**	1.282E+00
GSA	−3.862E+00	6.067E−04	−3.204E+00	4.201E−02	−6.517E+00	1.193E+00

Algorithm	F22		F23		F24	
	AVG	STD	AVG	STD	AVG	STD
CPA	−8.277E+00	2.648E+00	−8.373E+00	2.695E+00	**3.676E+05**	**2.006E+05**
PSO	−7.875E+00	8.814E−01	−8.148E+00	1.007E+00	7.374E+06	1.908E+06
DE	−1.040E+01	1.807E−15	−1.054E+01	1.616E−15	1.596E+07	4.997E+06
ABC	−1.040E+01	**3.299E−16**	−1.054E+01	1.582E−15	3.766E+06	1.535E+06
SCA	−3.951E+00	2.804E+00	−6.140E+00	2.222E+00	2.094E+08	7.040E+07
SSA	**−9.876E+00**	1.609E+00	−1.054E+01	1.096E−12	1.067E+06	4.225E+05
BA	−8.609E+00	2.074E+00	**−9.336E+00**	1.534E+00	6.156E+05	2.036E+05
FA	−8.824E+00	7.890E-01	−8.854E+00	8.084E−01	2.314E+08	4.509E+07
MFO	−8.076E+00	3.391E+00	−7.800E+00	3.704E+00	1.269E+08	1.543E+08
WOA	−1.040E+01	1.062E−07	−1.054E+01	**2.224E−07**	2.423E+07	1.115E+07
GWO	−1.040E+01	4.830E−07	−1.054E+01	4.154E−07	6.205E+07	5.313E+07
GSA	−7.245E+00	1.340E+00	−6.988E+00	1.215E+00	1.340E+06	3.315E+05

Algorithm	F25		F26		F27	
	AVG	STD	AVG	STD	AVG	STD
CPA	1.535E+04	1.213E+04	3.445E+02	3.178E+01	4.531E+02	3.600E+01
PSO	1.418E+08	1.160E+07	7.824E+02	6.845E+01	4.679E+02	3.958E+01
DE	6.111E+02	1.444E+03	**3.211E+02**	1.978E+01	4.819E+02	3.774E+01
ABC	**4.114E+02**	**2.065E+02**	9.500E+02	6.334E+02	**4.214E+02**	**2.895E+01**
SCA	1.481E+10	3.069E+09	3.652E+04	6.005E+03	1.318E+03	2.116E+02
SSA	1.367E+04	1.154E+04	3.751E+02	6.054E+01	4.723E+02	4.069E+01
BA	5.351E+05	3.265E+05	3.076E+02	**1.270E+01**	4.303E+02	4.287E+01
FA	1.473E+10	1.261E+09	5.965E+04	8.090E+03	1.520E+03	1.316E+02
MFO	1.226E+10	8.622E+09	1.128E+05	5.419E+04	1.575E+03	1.413E+03
WOA	8.920E+05	1.167E+06	2.577E+04	2.184E+04	5.662E+02	5.067E+01
GWO	2.474E+09	2.128E+09	2.923E+04	9.607E+03	6.344E+02	8.910E+01
GSA	2.020E+07	2.773E+06	7.236E+03	3.532E+03	4.598E+02	6.239E+01

Algorithm	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CPA	**5.200E+02**	**1.629E−04**	6.171E+02	3.542E+00	**7.000E+02**	3.145E−02
PSO	5.209E+02	5.256E−02	6.227E+02	2.204E+00	7.022E+02	1.224E−01
DE	5.205E+02	5.177E−02	6.176E+02	2.677E+00	**7.000E+02**	1.996E−04
ABC	5.202E+02	2.880E−02	**6.141E+02**	1.788E+00	**7.000E+02**	**5.517E−09**
SCA	5.209E+02	4.492E−02	6.334E+02	2.356E+00	8.308E+02	2.452E+01
SSA	5.201E+02	8.675E−02	6.175E+02	4.407E+00	**7.000E+02**	1.360E−02
BA	5.210E+02	4.440E−02	6.339E+02	2.557E+00	7.006E+02	2.036E−01
FA	5.209E+02	6.136E−02	6.332E+02	**1.101E+00**	8.278E+02	1.380E+01
MFO	5.203E+02	1.763E−01	6.249E+02	3.783E+00	8.315E+02	7.851E+01
WOA	5.203E+02	2.092E−01	6.351E+02	3.898E+00	7.008E+02	1.289E−01
GWO	5.209E+02	3.988E−02	6.130E+02	3.038E+00	7.159E+02	1.547E+01
GSA	5.209E+02	4.954E−02	6.074E+02	2.053E+00	7.012E+02	1.918E−02

Algorithm	F31		F32		F33	
	AVG	STD	AVG	STD	AVG	STD

CPA	**8.000E+02**	**2.704E−13**	1.018E+03	2.447E+01	1.001E+03	1.312E+00
PSO	9.669E+02	2.246E+01	1.101E+03	1.854E+01	5.009E+03	4.493E+02
DE	8.006E+02	6.744E-01	1.001E+03	**7.356E+00**	1.021E+03	2.113E+01
ABC	**8.000E+02**	**4.222E−14**	9.800E+02	1.082E+01	**1.000E+03**	**4.099E−02**
SCA	1.035E+03	2.005E+01	1.166E+03	1.653E+01	7.044E+03	5.401E+02
SSA	8.998E+02	3.009E+01	1.011E+03	3.124E+01	4.490E+03	5.899E+02
BA	1.024E+03	5.267E+01	1.183E+03	5.824E+01	5.323E+03	6.745E+02
FA	1.023E+03	9.813E+00	1.154E+03	1.127E+01	7.403E+03	2.741E+02
MFO	9.387E+02	4.607E+01	1.131E+03	5.231E+01	4.435E+03	8.969E+02
WOA	9.898E+02	4.339E+01	1.130E+03	6.215E+01	4.706E+03	7.236E+02
GWO	8.822E+02	1.641E+01	9.884E+02	1.553E+01	3.254E+03	4.993E+02
GSA	8.288E+02	4.527E+00	**9.538E+02**	1.094E+01	2.182E+03	2.591E+02
Algorithm	F34		F35		F36	
	AVG	STD	AVG	STD	AVG	STD
CPA	3.636E+03	5.443E+02	**1.200E+03**	5.158E−02	**1.300E+03**	1.340E−01
PSO	5.688E+03	5.053E+02	1.202E+03	2.961E−01	**1.300E+03**	6.230E−02
DE	5.472E+03	3.090E+02	1.201E+03	9.450E−02	**1.300E+03**	4.972E−02
AB	3.039E+03	**2.157E+02**	1.200E+03	**2.989E−02**	**1.300E+03**	2.532E−02
SCA	7.931E+03	3.709E+02	1.202E+03	2.384E−01	1.303E+03	2.845E−01
SSA	4.787E+03	8.046E+02	**1.200E+03**	1.995E−01	1.301E+03	**1.089E−01**
BA	5.527E+03	5.064E+02	1.201E+03	3.595E−01	1.301E+03	1.260E−01
FA	7.825E+03	2.862E+02	1.202E+03	2.438E−01	1.303E+03	2.081E−01
MFO	5.341E+03	8.799E+02	1.201E+03	2.318E−01	1.302E+03	1.221E+00
WOA	5.555E+03	6.909E+02	1.202E+03	3.954E−01	**1.300E+03**	9.840E−02
GWO	3.978E+03	5.592E+02	1.202E+03	1.063E+00	**1.300E+03**	5.914E−02
GSA	**2.771E+03**	3.121E+02	1.201E+03	8.975E−02	**1.300E+03**	1.705E−02
Algorithm	F37		F38		F39	
	AVG	STD	AVG	STD	AVG	STD
CPA	**1.400E+03**	1.023E-01	1.512E+03	6.941E+00	**1.610E+03**	6.793E−01
PSO	**1.400E+03**	1.412E−01	1.516E+03	1.839E+00	1.612E+03	4.881E−01
DE	**1.400E+03**	6.446E−02	1.511E+03	**6.473E−01**	1.611E+03	2.497E−01
ABC	**1.400E+03**	**1.373E-02**	1.507E+03	1.170E+00	**1.610E+03**	2.602E−01
SCA	1.442E+03	7.826E+00	3.961E+03	2.982E+03	1.613E+03	2.120E−01
SSA	**1.400E+03**	1.543E−01	**1.507E+03**	2.483E+00	1.611E+03	7.004E−01
BA	**1.400E+03**	1.180E−01	1.529E+03	8.778E+00	1.613E+03	2.613E−01
FA	1.439E+03	4.683E+00	1.259E+04	5.164E+03	1.613E+03	**1.799E−01**
MFO	1.433E+03	1.741E+01	2.410E+05	4.907E+05	1.613E+03	6.215E−01
WOA	**1.400E+03**	4.385E−02	1.569E+03	3.012E+01	1.613E+03	4.564E−01
GWO	1.405E+03	7.487E+00	1.740E+03	6.094E+02	1.611E+03	6.985E−01
GSA	**1.400E+03**	4.667E−02	1.513E+03	8.072E−01	1.613E+03	3.080E−01
Algorithm	F40		F41		F42	
	AVG	STD	AVG	STD	AVG	STD
CPA	**4.160E+04**	**1.816E+04**	3.910E+03	2.657E+03	1.910E+03	1.087E+01
PSO	2.432E+05	1.247E+05	1.793E+06	6.536E+05	1.918E+03	1.196E+01
DE	1.281E+06	5.998E+05	6.016E+03	2.668E+03	1.908E+03	**5.672E−01**
ABC	2.233E+06	5.998E+05	**2.228E+03**	**2.240E+02**	**1.907E+03**	8.408E-01
SCA	6.030E+06	2.568E+06	1.403E+08	7.701E+07	1.985E+03	1.824E+01
SSA	7.223E+04	5.276E+04	6.367E+03	5.054E+03	1.914E+03	2.056E+00

BA	6.341E+04	3.716E+04	8.157E+04	4.387E+04	1.920E+03	1.853E+01
FA	6.328E+06	1.858E+06	2.608E+08	6.795E+07	2.004E+03	1.392E+01
MFO	3.886E+06	6.208E+06	7.473E+07	3.174E+08	1.957E+03	4.990E+01
WOA	3.975E+06	1.973E+06	6.768E+03	4.935E+03	1.942E+03	3.425E+01
GWO	1.754E+06	2.386E+06	7.359E+06	2.367E+07	1.943E+03	2.654E+01
GSA	1.622E+05	1.090E+05	6.460E+04	2.087E+04	**1.907E+03**	6.565E-01
Algorithm	F43		F44		F45	
	AVG	STD	AVG	STD	AVG	STD
CPA	**2.067E+03**	**2.333E+01**	4.546E+04	2.757E+04	2.668E+03	1.615E+02
PSO	2.306E+03	8.243E+01	8.730E+04	4.722E+04	2.888E+03	2.124E+02
DE	3.960E+03	1.133E+03	2.277E+05	1.048E+05	**2.359E+03**	**8.172E+01**
ABC	7.245E+03	2.228E+03	1.901E+05	9.684E+04	2.451E+03	9.647E+01
SCA	1.363E+04	3.773E+03	1.062E+06	5.377E+05	2.935E+03	1.330E+02
SSA	2.301E+03	9.297E+01	5.301E+04	2.962E+04	2.586E+03	1.856E+02
BA	2.348E+03	8.760E+01	**4.187E+04**	**2.114E+04**	3.253E+03	2.515E+02
FA	1.896E+04	6.842E+03	1.577E+06	6.740E+05	2.914E+03	1.173E+02
MFO	7.073E+04	7.495E+04	8.311E+05	1.885E+06	2.999E+03	2.403E+02
WOA	2.365E+04	1.659E+04	8.344E+05	5.160E+05	2.945E+03	2.306E+02
GWO	1.684E+04	7.003E+03	3.185E+05	3.125E+05	2.521E+03	1.812E+02
GSA	3.018E+04	1.027E+04	9.441E+04	5.127E+04	3.106E+03	1.850E+02
Algorithm	F46		F47		F48	
	AVG	STD	AVG	STD	AVG	STD
CPA	**2.500E+03**	**0.000E+00**	**2.600E+03**	**0.000E+00**	**2.700E+03**	**0.000E+00**
PSO	2.616E+03	4.963E−01	2.625E+03	5.875E+00	2.713E+03	5.109E+00
DE	2.615E+03	1.388E−12	2.626E+03	3.348E+00	2.706E+03	8.967E−01
ABC	2.615E+03	1.412E-01	2.625E+03	4.099E+00	2.707E+03	1.074E+00
SCA	2.662E+03	1.186E+01	**2.600E+03**	3.539E−02	2.723E+03	7.575E+00
SSA	2.615E+03	2.895E-03	2.636E+03	6.940E+00	2.710E+03	3.060E+00
BA	2.615E+03	1.040E-03	2.672E+03	3.814E+01	2.731E+03	9.406E+00
FA	2.725E+03	1.847E+01	2.704E+03	4.337E+00	2.733E+03	4.208E+00
MFO	2.687E+03	5.763E+01	2.673E+03	3.493E+01	2.721E+03	1.060E+01
WOA	2.620E+03	3.321E+01	2.610E+03	2.405E+01	2.717E+03	1.470E+01
GWO	2.633E+03	8.369E+00	**2.600E+03**	5.069E-04	2.710E+03	4.898E+00
GSA	2.615E+03	5.088E+00	2.608E+03	4.604E−01	2.702E+03	1.348E−01
Algorithm	F49		F50		F51	
	AVG	STD	AVG	STD	AVG	STD
CPA	2.701E+03	1.453E-01	**2.900E+03**	**0.000E+00**	**3.000E+03**	**0.000E+00**
PSO	2.790E+03	3.055E+01	3.468E+03	2.768E+02	7.227E+03	9.782E+02
DE	**2.700E+03**	3.038E-02	3.182E+03	8.413E+01	3.630E+03	1.937E+01
ABC	**2.700E+03**	6.648E-02	3.106E+03	2.243E+00	3.762E+03	6.203E+01
SCA	2.702E+03	4.861E-01	3.360E+03	2.629E+02	4.714E+03	2.778E+02
SSA	2.700E+03	**1.015E-01**	3.351E+03	1.508E+02	3.840E+03	1.753E+02
BA	2.701E+03	1.337E-01	3.874E+03	4.408E+02	5.300E+03	6.919E+02
FA	2.702E+03	3.247E-01	3.787E+03	4.912E+01	4.227E+03	1.915E+02
MFO	2.706E+03	1.835E+01	3.615E+03	2.342E+02	3.966E+03	1.962E+02
WOA	2.704E+03	1.818E+01	3.694E+03	3.855E+02	4.874E+03	6.993E+02
GWO	2.741E+03	5.901E+01	3.329E+03	1.234E+02	3.940E+03	2.723E+02
GSA	2.771E+03	4.507E+01	3.214E+03	1.472E+02	4.571E+03	3.781E+02

Algorithm	F52		F53		+/−/=	Avg	Rank
	AVG	STD	AVG	STD			
CPA	**3.100E+03**	**0.000E+00**	**3.200E+03**	**0.000E+00**	~	**3.20**	**1**
PSO	6.624E+04	1.261E+05	1.115E+04	4.527E+03	48/2/3	9.21	10
DE	8.522E+03	2.019E+04	5.963E+03	9.032E+02	27/14/12	4.13	2
ABC	3.851E+03	8.538E+01	5.145E+03	3.938E+02	26/20/	4.34	3
SCA	9.763E+06	6.364E+06	2.048E+05	6.666E+04	51/0/2	9.85	11
SSA	1.922E+06	4.382E+06	9.973E+03	2.606E+03	35/3/1	5.80	4
BA	3.087E+07	3.391E+07	1.264E+04	8.667E+03	44/2/7	8.76	9
FA	2.610E+06	1.028E+06	1.618E+05	4.247E+04	50/0/3	11.35	12
MFO	3.518E+06	4.109E+06	5.656E+04	5.039E+04	46/0/7	8.51	8
WOA	5.810E+06	4.500E+06	7.142E+04	4.129E+04	44/1/8	7.21	6
GWO	3.878E+05	9.711E+05	5.055E+04	4.784E+04	41/4/8	6.75	5
GSA	4.758E+07	5.944E+07	7.193E+03	8.104E+02	45/5/3	8.13	7

F53 belong to CEC2014 functions. Of these, F26 is unimodal; F28, F35, and F39 are simple multimodal; F43 is a hybrid; and F50, F52, and F53 are composition. Fig. 7 shows CPA found the optimal solution on F1 and F4 early in the evaluation when other algorithms were just beginning to converge, suggesting that CPA is faster and more accurate than its competitors. Although CPA fell into local optimum on F26, it still ranked as the second-best algorithm. It can be observed from F10 and F20 that CPA found the optimal solution at a very fast speed in the early evaluation period, while some of the comparative algorithms fell into the local optimum. For F28, F39, and F43, CPA was still the first algorithm to find the optimal solution, although its convergence rate was slower. We can infer from the results that CPA's exploring and exploiting abilities are powerful, and the two stages have a good balance.

3.3 Parameter sensitivity analysis

Since an algorithm's parameters affect its convergence speed and accuracy, we analyzed the following parameters of CPA: Population size (N), the maximum number of evaluations (*MaxFEs*), parameter (w) and abandonment of the upper limit of probability. We set the abandonment probability upper limit to start from a when the abandonment probability upper limit was tested. The parameter of a was fixed to $w = 9$ for a total of eleven values. Similarly, in the analysis of a, we initialized the upper limit of probability to $2/3a$ and w to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, respectively. N and *MaxFEs* were set to 30 and 1000, respectively, and remained unchanged when the test abandoned the upper limit of probability. Each algorithm was tested on the 53 functions 30 times.

Table 3 provides a reference for comparing different values of the abandonment probability upper limit, which has a significant impact on an algorithm's performance. Specifically, the best performance was achieved when the abandonment of the probability upper limit was $2/3a$. Moreover, the maximum difference between the averages reached 3.7. Table 4 shows the influence of coefficient w is relatively small, and the maximum difference between the average is only about 1.2.

We used benchmark function F7 to test the influence of N and *MaxFEs* on CPA, where N was set to 5, 10, 30, 50, and 100, respectively. *MaxFEs* was initialized to 50, 100, 200, 500, and 1000. According to the test results in Fig. 8, the increase of N and *MaxFEs* improved the accuracy of CPA, but when they reached a certain level, the influence becomes minimal. Therefore, it is suggested to set these values according to specific needs in an experiment, because if the values are too large or too small, convergence will take a long time and results will not be ideal.

4 Comparative performance of CPA on engineering design problems

To further evaluate its efficiency, CPA was tested on various engineering design problems. The constraints included the death penalty and annealing, static,

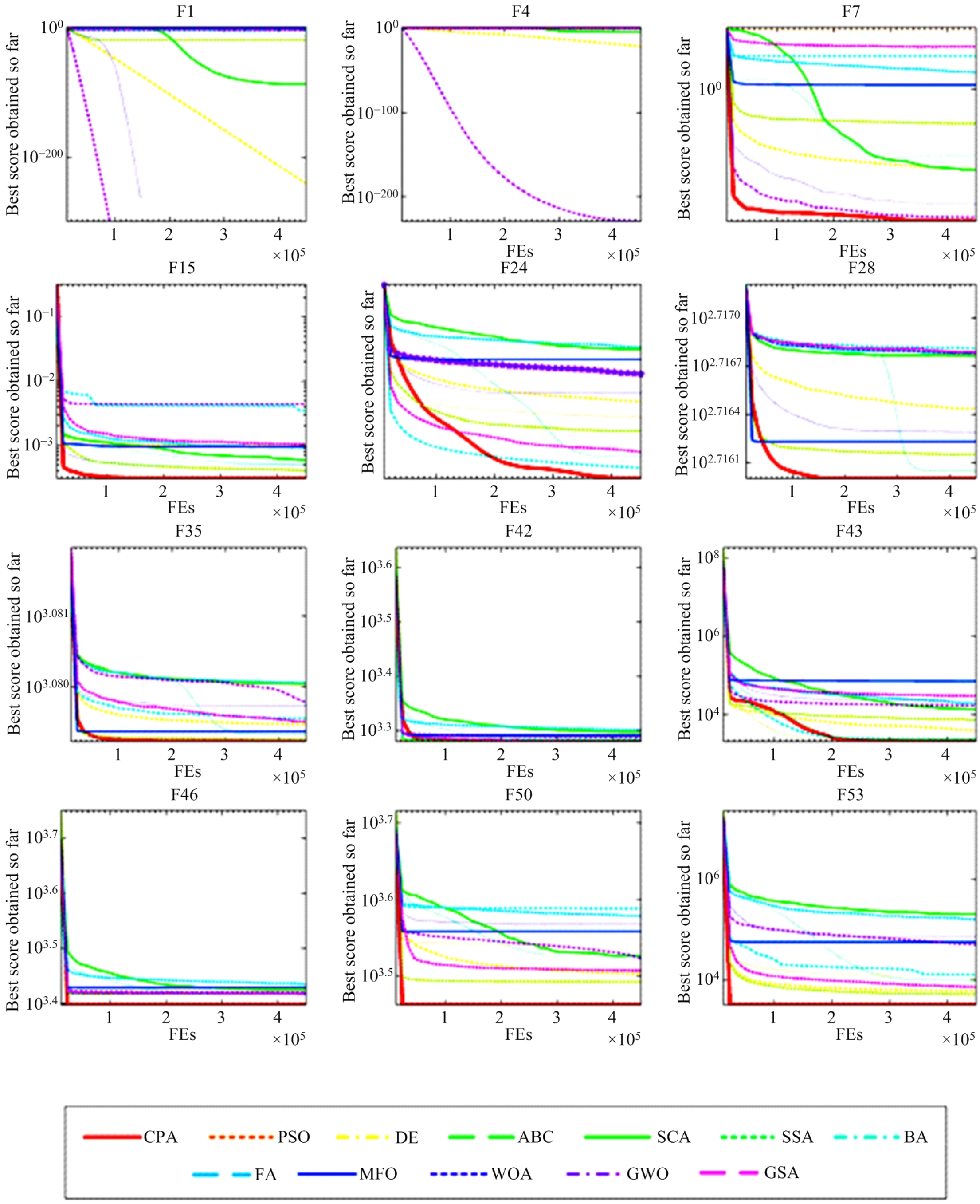


Fig. 6 Comparisons between CPA and traditional MAs.

dynamic, co-evolutionary, and adaptive behavior. To narrow down results, we selected the death penalty for comparisons in this work. The method assigns a large objective function value to them when searching individuals that violate any constraints. This method aims to automatically eliminate infeasible solutions in the heuristic algorithm's optimization, so it is unnecessary to calculate this scheme's infeasibility. The most prominent advantages of the death penalty are simplicity and low time consumption.

CPA was tested on the following five engineering constraints: Tension-compression spring, Welded beam, Pressure vessel, I-beam, Multiple disk clutch brake.

Table 2 Comparison results of CPA with the advanced algorithms on the 53 functions

Algorithm	F1		F2		F3	
	AVG	STD	AVG	STD	AVG	STD
CPA	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
m_SCA	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	2.69E−282	**0.000E+00**
OBSCA	1.25E−158	6.74E−158	1.831E−137	5.538E−137	5.076E−38	2.780E−37
MWOA	2.188E+04	7.718E+03	2.974E+05	1.042E+06	8.510E+04	3.019E+04
IWOA	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	5.047E+00	1.128E+01
CDLOBA	1.048E−04	3.073E−05	1.992E+01	8.798E+01	4.639E−04	1.845E−04
CBA	1.415E−08	6.143E−08	1.357E+03	7.331E+03	1.052E+01	5.026E+00
CGPSO	1.492E−09	2.176E−09	1.968E−05	1.915E−05	4.940E−02	4.483E−02
ALCPSO	2.514E−188	0.000E+00	1.377E−05	7.325E−05	1.946E−18	3.703E−18
DECLS	8.79E−119	1.32E−118	2.079E−70	1.745E−70	1.583E−05	7.473E−05

Algorithm	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CPA	**0.000E+00**	**0.000E+00**	2.145E+01	1.831E−01	1.551E−26	2.841E−26
m_SCA	5.14E−221	**0.000E+00**	2.679E+01	1.063E+00	2.344E+00	4.193E−01
OBSCA	1.060E−34	4.974E−34	2.771E+01	5.362E−01	3.844E+00	1.717E−01
MWOA	6.955E−01	2.374E+00	3.659E+07	1.940E+07	2.327E+04	7.569E+03
IWOA	1.010E−09	5.531E−09	2.279E+01	6.380E−01	8.517E−07	3.519E−07
CDLOBA	4.051E+01	7.625E+00	5.148E+01	7.075E+01	1.001E−04	3.649E−05
CBA	6.342E+00	7.897E+00	8.785E+01	1.163E+02	2.454E−08	1.232E−07
CGPSO	4.628E−06	4.775E−06	**1.701E−08**	**4.212E−08**	1.360E−09	1.673E−09
ALCPSO	3.215E−08	2.936E−08	3.205E+01	3.083E+01	**3.255E−30**	**1.661E−29**
DECLS	3.212E−11	1.773E−11	7.748E−09	1.262E−08	0.000E+00	0.000E+00

Algorithm	F7		F8		F9	
	AVG	STD	AVG	STD	AVG	STD
CPA	7.115E−05	1.705E−04	−1.257E+04	**5.189E−12**	**0.000E+00**	**0.000E+00**
m_SCA	4.159E−05	2.817E−05	−7.045E+03	6.606E+02	**0.000E+00**	**0.000E+00**
OBSCA	6.913E−04	3.587E−04	−4.200E+03	2.450E+02	**0.000E+00**	**0.000E+00**
MWOA	2.549E+01	2.048E+01	−5.421E+03	6.948E+02	3.047E+02	4.168E+01
IWOA	1.077E−04	1.085E−04	−1.221E+04	9.975E+02	**0.000E+00**	**0.000E+00**
CDLOBA	1.065E−02	1.053E−02	−7.179E+03	6.898E+02	1.298E+02	5.966E+01
CBA	1.206E−01	3.890E−01	−7.141E+03	7.509E+02	1.214E+02	4.671E+01
CGPSO	**7.411E−06**	**5.534E−06**	**−4.103E+04**	8.845E+03	2.141E−09	3.239E−09
ALCPSO	7.510E−02	2.667E−02	−1.172E+04	2.705E+02	1.636E+01	8.408E+00
DECLS	9.693E−06	1.044E−05	−1.257E+04	2.464E−09	1.660E−11	8.776E−11

Algorithm	F10		F11		F12	
	AVG	STD	AVG	STD	AVG	STD
CPA	**8.882E−16**	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.749E−29	1.276E−28
m_SCA	6.672E−01	3.655E+00	**0.000E+00**	**0.000E+00**	1.415E−01	4.864E−02
OBSCA	4.441E−15	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.913E−01	8.537E−02
MWOA	1.813E+01	1.282E+00	2.314E+02	5.463E+01	6.103E+07	6.883E+07
IWOA	2.783E−15	1.803E−15	**0.000E+00**	**0.000E+00**	1.396E−07	5.572E−08
CDLOBA	1.935E+01	9.355E−01	6.647E+01	6.601E+01	1.695E+01	8.470E+00
CBA	1.538E+01	3.469E+00	1.289E−02	1.177E−02	1.644E+01	5.412E+00
CGPSO	1.107E−05	1.342E−05	4.741E−09	8.832E−09	2.198E−11	2.468E−11

ALCPSO	1.135E+00	8.861E−01	2.602E−02	2.699E−02	1.227E−02	3.621E−02
DECLS	7.875E−15	6.486E−16	**0.000E+00**	**0.000E+00**	**1.571E−32**	**5.567E−48**

Algorithm	F13		F14		F15	
	AVG	STD	AVG	STD	AVG	STD
CPA	4.554E−28	7.038E−28	**9.980E−01**	1.487E−16	**3.075E−04**	**5.112E−17**
m_SCA	1.503E+00	2.505E−01	1.130E+00	5.034E−01	4.296E−04	3.166E−04
OBSCA	2.125E+00	1.176E−01	1.196E+00	6.054E−01	4.695E−04	1.609E−04
MWOA	1.415E+08	1.735E+08	5.326E+00	4.986E+00	2.792E−02	3.377E−02
IWOA	2.195E−06	1.160E−06	**9.980E−01**	9.823E−16	3.381E−04	1.672E−04
CDLOBA	3.824E+01	1.265E+01	1.989E+00	1.219E+00	3.331E−03	6.798E−03
CBA	2.863E+01	2.863E+01	1.856E+00	1.342E+00	7.041E−03	1.506E−02
CGPSO	2.732E−10	4.180E−10	1.064E+00	2.522E−01	6.631E−04	2.003E−04
ALCPSO	1.367E−02	5.440E−02	**9.980E−01**	**9.220E−17**	3.380E−04	1.672E−04
DECLS	**1.350E−32**	**5.567E−48**	1.034E+00	1.983E−01	4.445E−04	3.313E−04

Algorithm	F16		F17		F18	
	AVG	STD	AVG	STD	AVG	STD
CPA	**−1.032E+00**	**5.684E−16**	**3.979E−01**	**0.000E+00**	**3.000E+00**	**7.823E−16**
m_SCA	**−1.032E+00**	2.585E−10	**3.979E−01**	3.744E−09	**3.000E+00**	5.593E−08
OBSCA	**−1.032E+00**	2.722E−08	3.980E−01	6.649E−05	**3.000E+00**	6.572E−07
MWOA	−9.257E−01	1.847E−01	5.182E−01	2.130E−01	8.746E+00	1.076E+01
IWOA	**−1.032E+00**	9.749E−16	**3.979E−01**	1.018E−11	**3.000E+00**	3.780E−10
CDLOBA	**−1.032E+00**	1.318E−05	**3.979E−01**	8.478E−06	3.001E+00	1.366E−03
CBA	**−1.032E+00**	1.485E−06	**3.979E−01**	6.915E−07	**3.000E+00**	4.852E−05
CGPSO	**−1.031E+00**	1.913E−04	3.980E−01	1.001E−04	3.010E+00	1.180E−02
ALCPSO	**−1.032E+00**	5.904E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	2.109E−15
DECLS	**−1.032E+00**	6.775E−16	**3.979E−01**	**0.000E+00**	**3.000E+00**	1.195E−15

Algorithm	F19		F20		F21	
	AVG	STD	AVG	STD	AVG	STD
CPA	**−3.863E+00**	**2.542E−15**	−3.318E+00	2.171E−02	−7.774E+00	2.587E+00
m_SCA	**−3.863E+00**	1.832E−06	−3.240E+00	5.923E−02	−9.985E+00	9.224E−01
OBSCA	−3.862E+00	3.308E−04	−3.280E+00	1.890E−02	−9.869E+00	1.590E−01
MWOA	−3.735E+00	1.091E−01	−2.653E+00	4.343E−01	−3.343E+00	1.847E+00
IWOA	−3.862E+00	2.725E−03	−3.242E+00	1.005E−01	**−1.015E+01**	1.058E−07
CDLOBA	−3.859E+00	2.944E−03	−3.129E+00	4.376E−02	−6.529E+00	3.488E+00
CBA	**−3.863E+00**	5.284E−05	−3.245E+00	5.804E−02	−5.054E+00	3.042E+00
CGPSO	−3.860E+00	2.761E−03	−3.078E+00	1.061E−01	**−1.015E+01**	9.773E−11
ALCPSO	**−3.863E+00**	2.597E−15	−3.286E+00	5.542E−02	−8.547E+00	2.519E+00
DECLS	**−3.863E+00**	2.710E−15	**−3.322E+00**	**3.626E−05**	**−1.015E+01**	**7.227E−15**

Algorithm	F22		F23		F24	
	AVG	STD	AVG	STD	Avg	Rank
CPA	−8.100E+00	2.679E+00	−8.734E+00	2.593E+00	3.290E+05	**1.350E+05**
m_SCA	−9.876E+00	1.609E+00	−1.000E+01	1.636E+00	4.327E+07	2.832E+07
OBSCA	−1.003E+01	2.487E−01	−1.011E+01	2.475E−01	3.882E+08	1.112E+08
MWOA	−2.860E+00	1.649E+00	−3.080E+00	1.724E+00	1.857E+09	6.892E+08
IWOA	−1.040E+01	5.572E−08	−1.054E+01	5.190E−08	1.393E+07	8.157E+06
CDLOBA	−8.914E+00	2.958E+00	−8.115E+00	3.453E+00	**3.238E+05**	1.812E+05
CBA	−7.725E+00	3.629E+00	−6.424E+00	3.940E+00	3.991E+06	1.312E+06
CGPSO	**−1.040E+01**	1.201E−10	**−1.054E+01**	7.860E−11	8.169E+06	2.222E+06
ALCPSO	−1.005E+01	1.338E+00	−1.032E+01	9.933E−01	6.319E+06	5.585E+06

DECLS	**−1.040E+01**	**1.807E−15**	**−1.054E+01**	**1.649E−15**	2.248E+07	6.256E+06
Algorithm	F25		F26		F27	
	AVG	STD	AVG	STD	AVG	STD
CPA	1.766E+04	1.141E+04	**3.565E+02**	**3.827E+01**	**4.513E+02**	4.223E+01
m_SCA	5.467E+09	1.998E+09	2.245E+04	5.567E+03	8.127E+02	1.568E+02
OBSCA	2.280E+10	3.356E+09	4.783E+04	7.849E+03	2.223E+03	7.348E+02
MWOA	9.901E+10	2.208E+10	5.752E+05	5.535E+05	2.441E+04	1.004E+04
IWOA	4.516E+05	4.627E+05	1.306E+04	7.367E+03	5.586E+02	5.647E+01
CDLOBA	9.936E+03	1.068E+04	6.104E+04	2.018E+04	4.883E+02	5.822E+01
CBA	7.904E+04	3.741E+05	1.778E+03	4.473E+03	5.063E+02	2.889E+01
CGPSO	1.460E+08	2.219E+07	1.234E+03	1.957E+02	4.746E+02	4.260E+01
ALCPSO	1.912E+03	2.375E+03	3.779E+02	2.252E+02	5.273E+02	3.530E+01
DECLS	**7.740E+02**	**1.881E+03**	5.267E+02	4.877E+02	4.984E+02	**2.097E+01**
Algorithm	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CPA	**5.200E+02**	**2.243E−04**	**6.160E+02**	2.534E+00	**7.000E+02**	2.455E−02
m_SCA	5.206E+02	1.626E−01	6.206E+02	2.798E+00	7.467E+02	2.210E+01
OBSCA	5.209E+02	4.723E−02	6.316E+02	**1.393E+00**	8.844E+02	2.932E+01
MWOA	5.213E+02	6.794E−02	6.460E+02	2.910E+00	1.488E+03	1.820E+02
IWOA	5.202E+02	1.386E−01	6.295E+02	3.143E+00	**7.006E+02**	1.653E−01
CDLOBA	5.207E+02	3.399E−01	6.361E+02	3.370E+00	**7.000E+02**	9.591E−03
CBA	5.201E+02	1.762E−01	6.425E+02	2.854E+00	**7.000E+02**	1.545E−02
CGPSO	5.210E+02	5.258E−02	6.228E+02	3.066E+00	7.023E+02	1.555E−01
ALCPSO	5.208E+02	6.867E−02	6.164E+02	4.277E+00	**7.000E+02**	1.289E−02
DECLS	5.206E+02	4.869E−02	6.203E+02	1.488E+00	**7.000E+02**	**4.363E−07**
Algorithm	F31		F32		F33	
	AVG	STD	AVG	STD	AVG	STD
CPA	**8.001E+02**	**2.524E−01**	1.017E+03	2.203E+01	**1.002E+03**	**1.755E+00**
m_SCA	9.333E+02	2.818E+01	1.052E+03	2.381E+01	4.046E+03	6.226E+02
OBSCA	1.051E+03	1.785E+01	1.192E+03	1.630E+01	6.039E+03	3.769E+02
MWOA	1.239E+03	5.465E+01	1.383E+03	6.136E+01	9.545E+03	5.128E+02
IWOA	9.066E+02	1.714E+01	1.114E+03	5.163E+01	2.445E+03	3.908E+02
CDLOBA	1.049E+03	3.861E+01	1.229E+03	6.754E+01	5.347E+03	6.767E+02
CBA	1.008E+03	3.546E+01	1.172E+03	7.543E+01	5.558E+03	6.869E+02
CGPSO	9.805E+02	2.737E+01	1.119E+03	2.733E+01	5.669E+03	5.223E+02
ALCPSO	8.197E+02	1.079E+01	**9.980E+02**	2.808E+01	1.412E+03	2.965E+02
DECLS	8.011E+02	1.051E+00	1.015E+03	**8.373E+00**	1.031E+03	4.470E+01
Algorithm	F34		F35		F36	
	AVG	STD	AVG	STD	AVG	STD
CPA	**3.579E+03**	4.997E+02	**1.200E+03**	**4.286E−02**	1.301E+03	1.009E−01
m_SCA	4.610E+03	8.319E+02	1.201E+03	2.808E−01	1.301E+03	5.192E−01
OBSCA	7.075E+03	4.943E+02	1.202E+03	2.713E−01	1.303E+03	3.772E−01
MWOA	1.007E+04	4.826E+02	1.206E+03	9.982E−01	1.309E+03	1.134E+00
IWOA	5.427E+03	7.208E+02	1.201E+03	2.658E−01	1.301E+03	1.045E−01
CDLOBA	5.652E+03	7.017E+02	**1.200E+03**	2.214E−01	**1.300E+03**	1.134E−01
CBA	5.763E+03	8.245E+02	1.201E+03	4.547E−01	**1.300E+03**	1.406E−01
CGPSO	6.061E+03	4.943E+02	1.202E+03	3.129E−01	**1.300E+03**	9.512E−02
ALCPSO	4.098E+03	5.336E+02	1.201E+03	5.054E−01	1.301E+03	8.438E−02

−DECLS	5.779E+03	**3.601E+02**	1.201E+03	1.198E−01	**1.300E+03**	**5.850E−02**
Algorithm	F37		F38		F39	
	AVG	STD	AVG	STD	AVG	STD
CPA	**1.400E+03**	1.796E−01	1.513E+03	7.769E+00	**1.610E+03**	6.959E−01
m_SCA	1.411E+03	7.346E+00	2.214E+03	1.025E+03	1.611E+03	6.337E−01
OBSCA	1.460E+03	1.124E+01	1.426E+04	7.328E+03	1.613E+03	**1.389E−01**
MWOA	1.714E+03	5.629E+01	3.984E+06	4.264E+06	1.614E+03	2.657E−01
IWOA	**1.400E+03**	1.123E−01	1.554E+03	1.576E+01	1.612E+03	4.342E−01
CDLOBA	**1.400E+03**	1.741E−01	1.737E+03	8.671E+01	1.613E+03	4.509E−01
CBA	**1.400E+03**	1.188E−01	1.562E+03	1.798E+01	1.613E+03	3.217E−01
CGPSO	**1.400E+03**	1.339E−01	1.517E+03	1.235E+00	1.612E+03	3.149E−01
ALCPSO	1.401E+03	3.059E−01	**1.511E+03**	3.697E+00	1.612E+03	4.746E−01
DECLS	**1.400E+03**	**4.454E−02**	1.512E+03	**1.303E+00**	1.612E+03	2.415E−01
Algorithm	F40		F41		F42	
	AVG	STD	AVG	STD	AVG	STD
CPA	5.600E+04	2.684E+04	**5.088E+03**	**4.764E+03**	**1.908E+03**	1.339E+00
m_SCA	1.030E+06	5.682E+05	2.705E+07	3.517E+07	1.945E+03	2.463E+01
OBSCA	1.047E+07	4.398E+06	1.467E+08	9.625E+07	2.010E+03	1.183E+01
MWOA	2.360E+08	1.086E+08	6.119E+09	2.744E+09	2.659E+03	3.210E+02
IWOA	2.149E+06	1.481E+06	5.974E+03	3.608E+03	1.943E+03	4.172E+01
CDLOBA	**2.362E+04**	**1.855E+04**	7.921E+03	6.766E+03	1.972E+03	4.546E+01
CBA	2.023E+05	1.036E+05	1.049E+04	9.316E+03	1.928E+03	2.951E+01
CGPSO	2.761E+05	1.433E+05	1.953E+06	7.093E+05	1.917E+03	2.289E+00
ALCPSO	5.723E+05	6.545E+05	8.129E+03	6.473E+03	1.919E+03	2.420E+01
DECLS	1.533E+06	8.303E+05	1.020E+04	6.176E+03	**1.908E+03**	**7.177E−01**
Algorithm	F43		F44		F45	
	AVG	STD	AVG	STD	AVG	STD
CPA	**2.063E+03**	**2.713E+01**	4.903E+04	3.200E+04	2.710E+03	1.871E+02
m_SCA	9.629E+03	4.457E+03	5.113E+05	4.819E+05	2.561E+03	1.718E+02
OBSCA	2.527E+04	7.704E+03	1.953E+06	1.183E+06	3.132E+03	1.555E+02
MWOA	4.700E+06	5.323E+06	1.122E+08	8.287E+07	1.599E+04	2.983E+04
IWOA	1.147E+04	5.611E+03	7.312E+05	7.908E+05	2.893E+03	2.491E+02
CDLOBA	2.269E+04	1.259E+04	**2.837E+04**	**1.657E+04**	3.423E+03	3.906E+02
CBA	2.555E+03	3.179E+02	7.348E+04	4.454E+04	3.430E+03	3.740E+02
CGPSO	2.343E+03	5.773E+01	1.264E+05	7.996E+04	2.855E+03	2.061E+02
ALCPSO	2.556E+03	2.462E+02	5.603E+04	4.688E+04	2.662E+03	1.771E+02
DECLS	4.568E+03	1.639E+03	3.425E+05	1.514E+05	**2.387E+03**	**8.056E+01**
Algorithm	F46		F47		F48	
	AVG	STD	AVG	STD	AVG	STD
CPA	**2.500E+03**	**0.000E+00**	**2.600E+03**	**0.000E+00**	**2.700E+03**	**0.000E+00**
m_SCA	2.639E+03	9.903E+00	**2.600E+03**	6.768E−04	2.714E+03	4.213E+00
OBSCA	2.680E+03	1.589E+01	**2.600E+03**	1.469E−04	**2.700E+03**	5.278E−04
MWOA	3.757E+03	3.789E+02	2.808E+03	3.914E+01	2.828E+03	4.214E+01
IWOA	2.618E+03	1.739E+00	2.602E+03	1.354E+00	2.715E+03	1.430E+01
CDLOBA	2.615E+03	3.789E−01	2.713E+03	5.314E+01	2.722E+03	1.233E+01
CBA	2.616E+03	1.430E−01	2.682E+03	3.643E+01	2.737E+03	1.336E+01
CGPSO	**2.500E+03**	9.389E−04	**2.600E+03**	9.499E−03	**2.700E+03**	1.505E−05
ALCPSO	2.615E+03	5.937E−02	2.639E+03	7.224E+00	2.711E+03	4.146E+00

DECLS	**2.500E+03**	6.901E−04	**2.600E+03**	7.578E−03	**2.700E+03**	7.747E−06

Algorithm	F49		F50		F51	
	AVG	STD	AVG	STD	AVG	STD
CPA	2.701E+03	1.447E−01	**2.900E+03**	**0.000E+00**	**3.000E+03**	**0.000E+00**
m_SCA	2.701E+03	3.736E−01	3.151E+03	6.149E+01	3.876E+03	1.196E+02
OBSCA	2.704E+03	5.210E−01	3.224E+03	4.059E+01	5.456E+03	3.081E+02
MWOA	2.764E+03	8.949E+01	4.408E+03	2.019E+02	8.187E+03	1.082E+03
IWOA	2.701E+03	1.489E−01	3.582E+03	3.113E+02	4.689E+03	5.042E+02
CDLOBA	2.704E+03	1.814E+01	3.727E+03	4.539E+02	5.456E+03	7.152E+02
CBA	2.722E+03	8.145E+01	3.969E+03	4.524E+02	5.536E+03	6.722E+02
CGPSO	2.800E+03	1.346E−06	3.145E+03	3.610E+02	**3.000E+03**	2.588E−02
ALCPSO	2.755E+03	6.117E+01	3.452E+03	2.301E+02	4.472E+03	5.191E+02
DECLS	**2.700E+03**	**5.098E−02**	3.047E+03	1.913E+02	3.021E+03	1.176E+02

Algorithm	F52		F53				
	AVG	STD	AVG	STD	+/−/=	Avg	Rank
CPA	**3.100E+03**	**0.000E+00**	**3.200E+03**	**0.000E+00**	~	**2.88**	**1**
m_SCA	8.126E+05	2.707E+06	4.283E+04	1.997E+04	44/2/7	6.11	6
OBSCA	1.209E+07	6.124E+06	3.712E+05	1.448E+05	47/1/5	7.97	9
MWOA	2.509E+08	1.549E+08	5.695E+06	3.390E+06	53/0/0	10.76	10
IWOA	5.345E+06	4.456E+06	2.079E+04	1.007E+04	42/2/9	5.68	4
CDLOBA	1.741E+07	1.711E+07	1.314E+05	2.196E+05	42/5/6	7.63	8
CBA	4.239E+07	4.060E+07	6.782E+04	1.472E+05	47/1/5	7.25	7
CGPSO	4.138E+03	9.991E+02	1.009E+04	9.652E+03	44/6/3	6.11	5
ALCPSO	3.452E+06	1.169E+07	1.130E+04	6.340E+03	33/6/14	4.86	3
DECLS	4.911E+03	8.387E+02	6.763E+03	8.839E+02	27/14/12	3.72	2

4. 1 Tension-compression spring problem

The mathematical model of a Tension/Compression Spring (TCS) design is designed to minimize the weight of the spring. This problem requires the optimization of three design variables, including the wire diameter (*d*), mean coil diameter (*D*), and number of active coils (*N*). Fig. 9 illustrates this problem.

The mathematical model of this problem is as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$

Objective function:

$$f(x)_{\min} = x_1^2 x_2 x_3 + 2x_1^2 x_2$$

Subject to

$$h_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$h_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5180 x_1^2} \le 0$$

$$h_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \le 0$$

$$h_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variable ranges:

$$0.05 \le x_1 \le 1.00$$

$$0.25 \le x_2 \le 1.30$$

$$2.00 \le x_3 \le 15.0$$

The model has been solved by mathematical optimization tools and metaheuristic methods. For instance, Coello *et al*. realized this problem by using Genetic Algorithms (GA) and obtained an adaptation value of 0.0127048. The experimental results, after testing many algorithms, show that the optimal weight can reach 0.0126650.

Table 5 reveals that the weight calculated by CPA is smaller than the current optimal weight, which also proves the superiority of the CPA in TCSD.

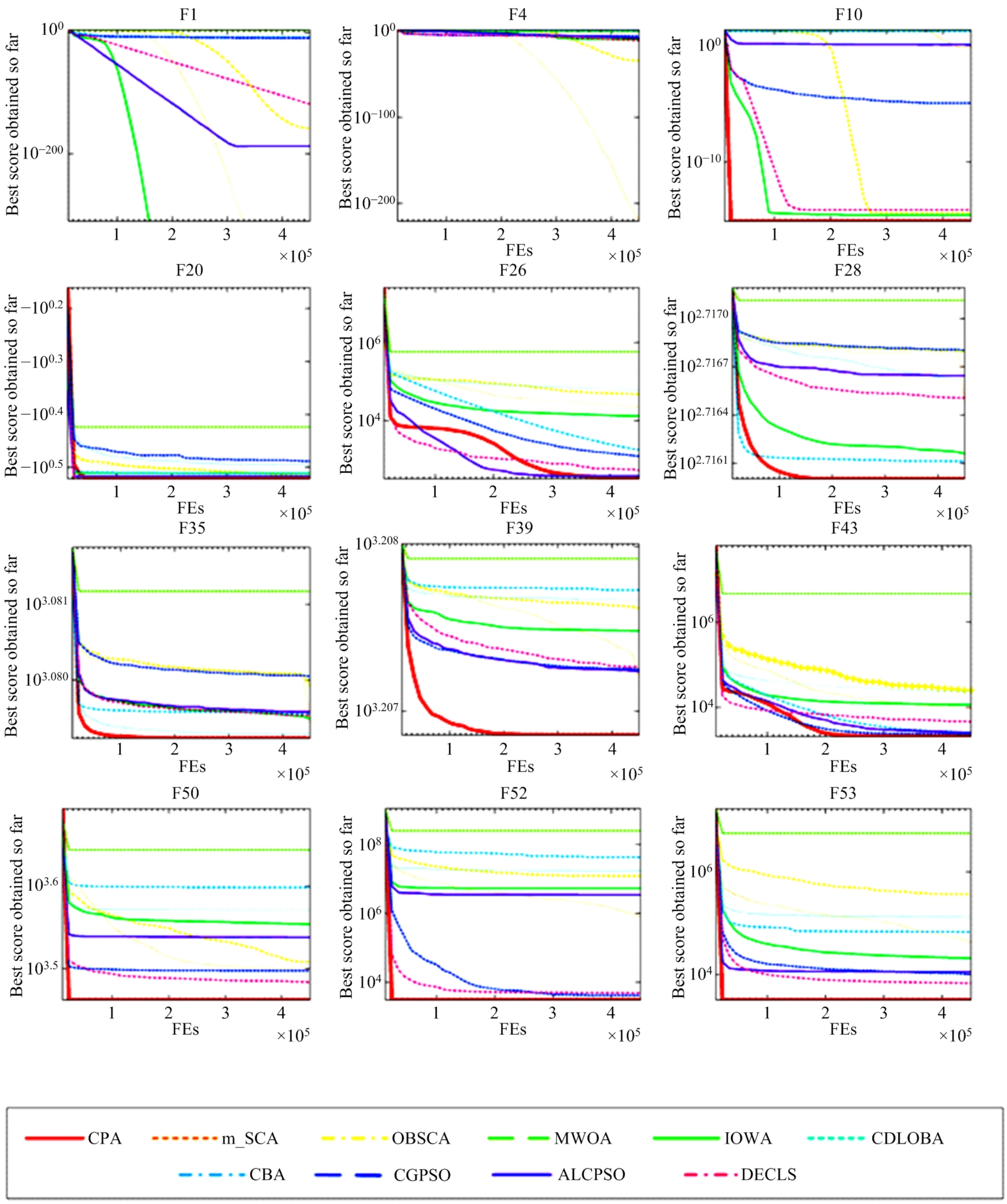


Fig. 7 Comparisons between CPA and advanced MAs.

4. 2 Welded beam design problem

The purpose of this problem is to find the lowest cost of welded beams under four constraints; shear stress (τ), bending stress (θ), buckling load (P_C), and deflection (δ). These four variables of this problem include the welding seam thickness (h), welding joint length, beam width (t), and beam thickness (b). Fig. 10 demonstrates the elements of this problem.

The mathematical model is as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [h\ l\ t\ b]$

Minimize

$$f(x)_{\min} = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_4)$$

Subject to $g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0$

Table 3 Ranking results of upper limits for different *S*

Fun	*a*	2*a*/3	*a*/2	*a*/3	*a*/4	*a*/5	*a*/6	*a*/7	*a*/8	*a*/9	*a*/10
ARV	6.7358	3.00	3.8490	4.5094	4.9811	4.8490	4.7547	5.9622	6.0566	6.00	6.1509
Rank	11	1	2	3	6	5	4	7	9	8	10

Table 4 Ranking of results with different values of parameter *w*

Fun	*w* =1	*w* =2	*w* =3	*w* =4	*w* =5	*w* =6	*w* =7	*w* =8	*w* =9	*w* =10
ARV	6.30	6.00	5.71	5.56	5.45	5.33	5.18	5.20	5.06	5.13
Rank	10	9	8	7	6	5	4	3	1	2

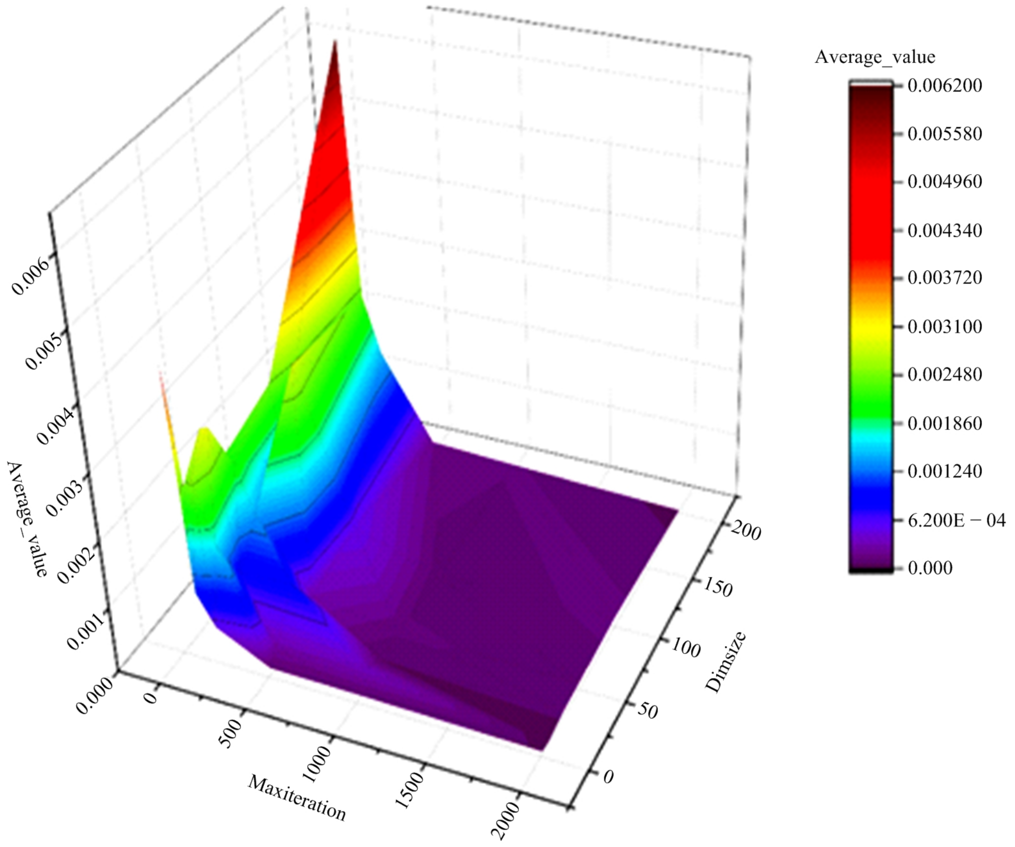


Fig. 8 The influence of *N* and Max_iter.

$$g_2(\overset{r}{x}) = \sigma(\overset{r}{x}) - \sigma_{\max} \le 0$$

$$g_3(\overset{r}{x}) = \delta(\overset{r}{x}) - \delta_{\max} \le 0$$

$$g_4(\overset{r}{x}) = x_1 - x_4 \le 0$$

$$g_5(\overset{r}{x}) = P - P_c(\overset{r}{x}) \le 0$$

$$g_6(\overset{r}{x}) = 0.125 - x_1 \le 0$$

$$g_7(\overset{r}{x}) = 1.10471 x_1^{\ 2} + 0.04811 x_3 x_4 (14.0 + x_4) - 5.0 \le 0$$

Variable range $0.1 \le x_1 \le 2,\ 0.1 \le x_2 \le 10,\ 0.1 \le x_3 \le 10,$

$0.1 \le x_4 \le 2$

where $\tau(\overset{r}{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$ $\quad \tau' = \frac{P}{\sqrt{2x_1x_2}}$

$$\tau'' = \frac{MR}{J} \quad M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$

$$J = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2]\}$$

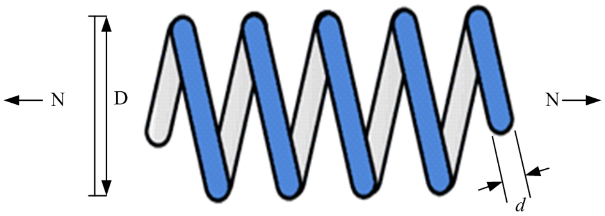


Fig. 9 The structure of the tension-compression spring.

Table 5 Results of CPA versus other algorithms for TCS design problem

Algorithm	Optimal values for variables			Optimum cost
	d	*D*	*N*	
CPA	**0.051741**	**0.357978**	**11.215480**	**0.0126650**
GA	0.051480	0.351661	11.632201	0.0127048
RO[86]	0.051370	0.349096	11.762790	0.0126788
IHS[87]	0.051154	0.349871	12.076432	0.0126706
Constraint correction method (Arora)[52]	0.050000	0.315900	14.250000	0.0128334
Mathematical optimization method (Belegundu)[52]	0.053396	0.399180	901854000	0.0127303
WOA	0.051207	0.345215	12.0043032	0.0126763
PSO	0.015728	0.357644	11.244543	0.0126747

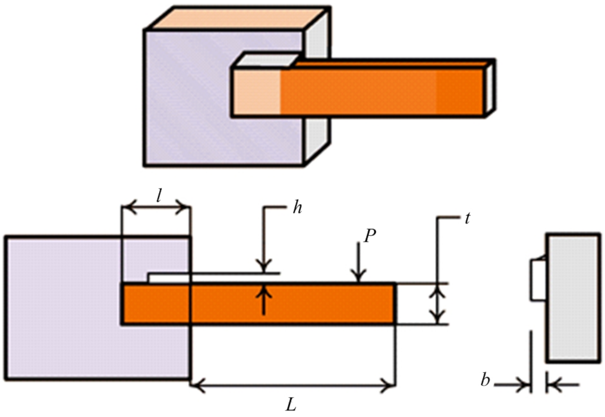


Fig. 10 The structure of a welded beam.

Table 6 Results compared with other approaches

Algorithm	Optimal values for variables				Optimum cost
	h	*l*	*t*	*b*	
CPA	**0.188500**	**3.562000**	**9.134835**	**0.205245**	**1.723916**
RO	0.203687	3.528467	9.004233	0.207241	1.735344
SSA	0.205700	3.471400	9.036600	0.205700	1.724910
CDE	0.203137	3.542998	9.033498	0.206179	1.733462
GWO	0.205700	3.478400	9.036800	0.205800	1.726240
WOA	0.175500	3.893500	9.018932	0.206538	1.736036
GSA	0.182100	3.857000	10.00000	0.202400	1.879950

$$\sigma(\overset{r}{x})=\frac{6PL}{x_4x_3^2},\ \delta(\overset{r}{x})=\frac{6PL^3}{Ex_4x_3^2}$$

$$P_C(\overset{r}{x})=\frac{4.013E\sqrt{\frac{{x_4}^6x_3^2}{36}}}{L^2}(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}})$$

$$P_C=60001b,\ L=14,\ \delta_{\max}=0.25$$

$$E=30\times1^6\,psi,\ G=12\times10^6\,psi$$

$$\tau_{\max}=13600\,psi,\ \sigma_{\max}=30000\,psi$$

The performance of CPA was compared with those of RO, SSA, CDE[88], GWO, WOA, and GSA are compared for this problem. The current optimal weight is 1.72452. We can find from Table 6 that the SSA has the best performance, achieving a weight of 1.723916.

4.3 Pressure vessel design problem

This problem is considered a well-studied structural optimization test case, which aims to minimize the production cost of cylindrical pressure vessels. One end of the container is covered, and the other end is a hemisphere, for which the total manufacturing costs is determined by four variables – the thickness of the shell T_s, the thickness of the head (T_h), inner radius (R), and the span of the cylindrical part of the container (L) – determine the total manufacturing cost. Fig. 11 contains an illustration of this problem.

The relevant formulas are as follows:

Consider $\overset{r}{x}=[x_1\ x_2\ x_3\ x_4]=[T_s\ T_h\ R\ L]$

Minimize

$$f(\overset{r}{x})_{\min}=0.6224x_1x_3x_4+1.7781x_3{x_1}^2+3.1661x_4{x_1}^2+19.84x_3{x_1}^2$$

Subject to $g_1(\overset{r}{x})=-x_1+0.0193x_3\le0$

$g_2(\overset{r}{x})=-x_2+0.0095x_3\le0$

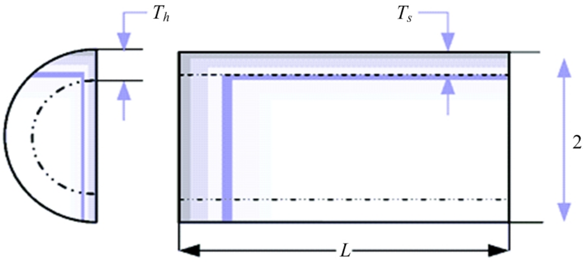


Fig. 11 The structure of the pressure vessel.

Table 7 Results of pressure vessel design problem compared with other methods

Algorithm	Optimal values for variables				Optimum cost
	T_S	T_h	R	L	
CPA	0.81250	0.437500	42.088230	176.763300	6060.9590
PSO	0.812500	0.437500	42.091266	176.746500	6061.0777
MFO	**0.812500**	**0.437500**	**42.098400**	**176.636600**	**6059.7143**
GWO	0.812500	0.434500	42.089200	176.758700	6051.5639
WOA	0.812500	0.437500	42.098300	176.639000	6059.7410
BA	**0.812500**	**0.437500**	**42.098445**	**176.636595**	**6059.7143**
CPSO	0.812500	0.437500	42.091266	176.746500	6061.0777
HPSO	**0.812500**	**0.437500**	**42.098400**	**176.636600**	**6059.7143**
Lagrangian multiplier	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch-bound	1.125000	0.625000	47.700000	117.701000	8129.1036

Table 8 Results of I-beam design problem compared with other methods

Algorithm	Optimal values for variables				Optimum cost
	B	h	T_w	T_f	
CPA	49.9141	79.8044	0.935684	2.300281	0.0132240
ARSM	48.4200	79.9900	0.900000	2.400000	0.0157000
IARSM	48.4200	79.9900	0.900000	2.400000	0.1310000
CS	50.0000	80.0000	0.900000	2.321675	0.0130747
SOS	**50.0000**	**80.0000**	**0.900000**	**2.321790**	**0.0130741**

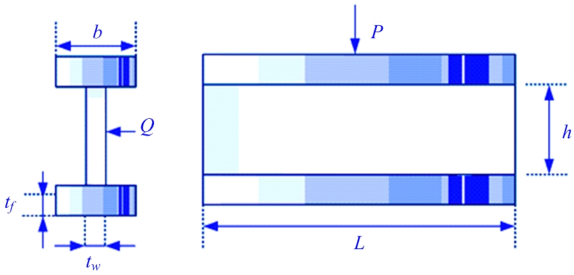


Fig. 12 The structure of I-beam.

$$g_3(\vec{x}) = -\pi x_4 x_3^2 + \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(\vec{x}) = x_4 - 240 \le 0$$

Variable range

$$0 \le x_1 \le 99,\ 0 \le x_2 \le 99,\ 10 \le x_3 \le 200,\ 10 \le x_4 \le 200$$

For this engineering problem, this work compared the performance of CPA with those of PSO, MFO, GWO, WOA, BA, CPSO[89], HPSO[90], Lagrangian multiplier[91] and Branch-bound[92]. Table 7 shows that the optimal value of the CPA was about 1.2 higher than those of MFO, BA, and HPSO. Thus, the latter three MAs achieved minimum optimal consumption, but still exceeded the current optimal weight of 6059.7143.

4. 4 I-beam design problem

The purpose of this problem is to minimize the vertical deflection of the I-beams. Related parameters include the length, height, and two thicknesses. Fig. 12 shows the configuration of this problem.

The mathematical model of the problem is as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [b\ h\ t_w\ t_f]$

Objective

$$f(\vec{x})_{\min} = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2}$$

Subject to $g_1(\vec{x}) = 2bt_w + t_w(h-2t_f) \le 0$

$$g_1(\vec{x}) = \frac{18h\times 10^4}{t_w(h-2t_f)^3 + 2bt_f(4t_f + 3h(h-2t_f))} + \frac{15b\times 10^3}{(h-2t_f)t_w^3 + 2t_f b^3}$$

Variable range $10 \le x_1 \le 50$

$$10 \le x_2 \le 80$$

$$0.9 \le x_3 \le 5$$

$$0.9 \le x_4 \le 5$$

Table 8 is a comparison between CPA and

Table 9 Results of Multiple disk clutch brake compared with other methods

Algorithm	r_i	r_0	t	F	Z	Optimal cost
CPA	**70.00000**	**90.00000**	**1**	**99000**	**3.00000**	**0.313656**
WCA	**70.00000**	**90.00000**	**1**	**910.0000**	**3.000000**	**0.313656**
PVS	70.00000	90.00000	1	980.0000	3.000000	0.313660
TLBO	**70.00000**	**90.00000**	**1**	**810.0000**	**3.000000**	**0.313656**

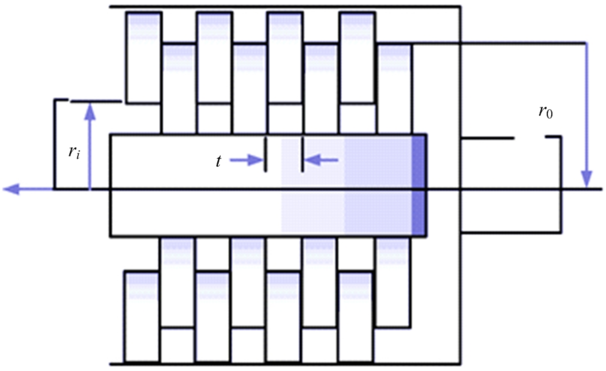


Fig. 13 The structure of multiple disk clutch brake.

ARSM[93], IARSM[93], CS, and SOS[94] on I-beam problems. We can find from Table 8 that CPA can help the vertical deflection of the I-beam to minimize more than the other four algorithms, which is the same as the current optimal value of 0.006626, which shows that CPA is more suitable for engineering problems.

4. 5 Multiple disk clutch brake problem

The objective of this minimization and discrete optimization problem is to use five discrete design variables to minimize the quality of multiple disc clutch brakes. The five variables are the actuating force, inner and outer radii, number of 27 friction surfaces, and thickness of discs. Fig. 13 shows the configuration of this problem.

The mathematical model for this problem is as follows:

$$f(\overset{\mathrm{r}}{x})_{\min} = \pi(r_0^2 - r_i^2)t(Z+1)\rho$$

subject to:

$$g_1(\overset{\mathrm{r}}{x}) = r_0 - r_i - \Delta r \geq 0$$

$$g_2(\overset{\mathrm{r}}{x}) = l_{\max} - (Z+1)(t+\delta) \geq 0$$

$$g_3(\overset{\mathrm{r}}{x}) = P_{\max} - P_{rz} \geq 0$$

$$g_4(\overset{\mathrm{r}}{x}) = P_{\max} v_{sr_{\max}} - P_{rz} v_{sr} \geq 0$$

$$g_5(\overset{\mathrm{r}}{x}) = v_{sr_{\max}} - v_{sr} \geq 0$$

$$g_6(\overset{\mathrm{r}}{x}) = T_{\max} - T \geq 0$$

$$g_7(\overset{\mathrm{r}}{x}) = M_h - sM_s \geq 0$$

$$g_8(\overset{\mathrm{r}}{x}) = T \geq 0$$

$$M_h = \frac{2}{3}\mu FZ \frac{r_0^3 - r_i^3}{r_0^2 - r_i^3}\ P_{rz} = \frac{F}{\pi(r_0^2 - r_i^2)}$$

$$v_{sr} = \frac{2\pi n(r_0^3 - r_i^3)}{90(r_0^2 - r_i^2)}\ T = \frac{I_Z \pi n}{30(M_h + M_f)}$$

$$\Delta r = 20\text{mm}, I_Z = 55\text{kgmm}^2, P_{\max} = 1\text{MPa},$$
$$F_{\max} = 1000\text{N}, T_{\max} = 15\text{s}, \mu = 0.5,$$

$$s = 1.5, M_s = 40\text{Nm}, M_f = 3\text{Nm}, n = 250\,\text{rpm},$$
$$v_{sr_{\max}} = 10\text{m}\cdot\text{s}^{-1}, l_{\max} = 30\text{mm},$$

$$r_{i_{\min}} = 60, r_{i_{\max}} = 80, r_{0_{\min}} = 90, r_{0_{\max}} = 110,$$
$$t_{\min} = 1.5, t_{\max} = 3, F_{\min} = 600,$$

$$F_{\max} = 1000, Z_{\min} = 2, Z_{\max} = 9.$$

This work compared CPA with WCA[95], PVS[96], and TLBO to minimize the quality of multiple disc clutch brakes. Table 9 shows that the quality of CPA, reaching 0.313656, is equal to or less than that of other algorithms. This indicates that the proposed algorithm has a stronger optimization ability and can find more high-quality problem solutions.

5 Conclusions and future work

This paper proposes a metaheuristic algorithm, named the Colony Predation Algorithm (CPA), inspired by social animals' characteristics to solve the optimization problem. The algorithm achieves the correct balance between exploitation and exploration and can quickly converge in the early and middle stages without falling into Local Optimization (LO). The algorithm motivated by the common foraging characteristics of social animals follows the mechanism based on the idea of selective abandonment to simulate the impact of this strategy on individual activities.

This study qualitatively analyzed the algorithm according to the four-index search history, first-dimensional trajectory, average fitness, and convergence curve. The algorithm's performance was evaluated on benchmark and CE2014 functions, and the Friedman test and Wilcoxon test were used for statistical evaluation. The experimental results show that the algorithm has a strong search ability to find the target solution space quicker and subsequently perform exploitation more effectively than other algorithms. The CPA can also identify the optimal solution faster and better on complex multimode functions and exhibits a stronger exploitation ability on complex functions.

Simultaneously, to verify its applicability in practical problems, CPA was applied to tension-compression spring, welded beam, pressure vessel, and other engineering problems. The experimental results show reveal that CPA's average is about 0.9 lower than the second DE and about 0.84 lower than the second improved DECLS. Thus, the CPA can achieve the optimization of production engineering problems and significantly reduce manufacturing costs.

This work further streamlined the CPA to make it easier to add new mechanisms. In future research, CPA can be applied for parameter optimization, binary feature selection, and image segmentation of machine learning algorithms and combined with machine learning to predict disease for disease prediction.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (62076185, U1809209). We acknowledge the efforts of Ali Asghar Heidari (https://aliasgharheidari.com) during the preparation and development of this research. We also thank the comments of reviewers and editor.

References

[1] Huang H, Feng X A, Zhou S Y, Jiang J H, Chen H L, Li Y P, Li C Y. A new fruit fly optimization algorithm enhanced support vector machine for diagnosis of breast cancer based on high-level features. *Bmc Bioinformatics*, 2019, **20**, https://doi.org/10.1186/s12859-019-2771-z.

[2] Cao B, Zhao J W, Gu Y, Ling Y B, Ma X L. Applying graph-based differential grouping for multiobjective large-scale optimization. *Swarm and Evolutionary Computation*, 2020, **53**, 100626.

[3] Fu X W, Pace P, Aloi G, Yang L, Fortino G. Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm. *Computer Networks*, 2020, 107327.

[4] Cao B, Zhao J W, Yang P, Gu Y, Muhammad K, Rodrigues J J, De Albuquerque V H C. Multiobjective 3-D topology optimization of next-generation wireless data center network. *IEEE Transactions on Industrial Informatics*, 2019, **16**, 3597–3605.

[5] Cao B, Dong W N, Lv Z H, Gu Y, Singh S, Kumar P. Hybrid microgrid many-objective sizing optimization with fuzzy decision. *IEEE Transactions on Fuzzy Systems*, 2020.

[6] Zeng G Q, Lu Y Z, Mao W J. Modified extremal optimization for the hard maximum satisfiability problem. *Journal of Zhejiang University SCIENCE C*, 2011, **12**, 589–596.

[7] Zeng G Q, Chen J, Dai Y X, Li L M, Zheng C W, Chen M R. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing*, 2015, **160**, 173–184.

[8] Gupta S, Deep K, Heidari A A, Moayedi H, Chen H L. Harmonized salp chain-built optimization. *Engineering with Computers*, 2019, https://doi.org/10.1007/s00366-019-

00871-5.

[9] Zhang H L, Cai Z N, Ye X J, Wang M J, Kuang F J, Chen H L, Li C Y, Li Y P. A multi-strategy enhanced salp swarm algorithm for global optimization. *Engineering with Computers*, 2020, https://doi.org/10.1007/s00366-020-01099-4.

[10] Xu Y T, Chen H L, Luo J, Zhang Q, Jiao S, Zhang X Q. Enhanced Moth-flame optimizer with mutation strategy for global optimization. *Information Sciences*, 2019, **492**, 181–203.

[11] Heidari A A, Ali Abbaspour R, Chen H L. Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. *Applied Soft Computing*, 2019, **81**, 105521.

[12] Song S M, Wang P J, Heidari A A, Wang M J, Zhao X H, Chen H L, He W M, Xu S L. Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns. *Knowledge-Based Systems*, 2020, https://doi.org/10.1016/j.knosys.2020.106425.

[13] Zhang Y N, Liu R J, Wang X, Chen H L, Li C Y. Boosted binary Harris hawks optimizer and feature selection. *Engineering with Computers*, 2020. https://doi.org/10.1007/s00366-020-01028-5.

[14] Xu X, Chen H L. Adaptive computational chemotaxis based on field in bacterial foraging optimization. *Soft Computing*, 2014, **18**, 797–807.

[15] Fan Y, Wang P J, Mafarja M, Wang M J, Zhao X H, Chen H L. A bioinformatic variant fruit fly optimizer for tackling optimization problems. *Knowledge-Based Systems*, 2021, **213**, 106704.

[16] Yu H L, Li W S, Chen C C, Liang J, Gui W Y, Wang M J, Chen H L. Dynamic Gaussian bare-bones fruit fly optimizers with abandonment mechanism: Method and analysis. *Engineering with Computers*, 2020, https://doi.org/10.1007/s00366-020-01174-w.

[17] Wang X Y, Chen H L, Heidari A A, Zhang X, Xu J, Xu Y T, Huang H. Multi-population following behavior-driven fruit fly optimization: A Markov chain convergence proof and comprehensive analysis. *Knowledge-Based Systems*, 2020, **210**, 106437.

[18] Ding L, Li S, Gao H B, Chen C, Deng Z Q. Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, **50**, 2512–2523.

[19] Wang J H, Zhu P S, He B T, Deng G Y, Zhang C L, Huang X. An adaptive neural sliding mode control with ESO for uncertain nonlinear systems. *International Journal of Control, Automation and Systems*, 2020.

[20] Hu J W, Wang M, Zhao C H, Pan Q, Du C. Formation control and collision avoidance for multi-UAV systems based on Voronoi partition. *Science China Technological Sciences*, 2020. **63**, 65–72.

[21] Qian J M, Feng S J, Tao T Y, Hu Y, Li Y X, Chen Q, Zuo C. Deep-learning-enabled geometric constraints and phase unwrapping for single-shot absolute 3d shape measurement. *APL Photonics*, 2020, **5**, 046105.

[22] Qian J M, Feng S J, Li Y X, Tao T Y, Han J, Chen Q, Zuo C. Single-shot absolute 3D shape measurement with deep-learning-based color fringe projection profilometry. *Optics Letters*, 2020, **45**, 1842–1845.

[23] Huang Y C, Wang J H, Wang F, He B T. Event-triggered adaptive finite-time tracking control for full state constraints nonlinear systems with parameter uncertainties and given transient performance. *ISA Transactions*, 2021, **108**, 131–143.

[24] Chen Z C, Wang J H, Ma K M, Huang X, Wang T. Fuzzy adaptive two - bits - triggered control for nonlinear uncertain system with input saturation and output constraint. *International Journal of Adaptive Control and Signal Processing*, 2020, **34**, 543–559.

[25] Shida H, Fei G, Quan Z, Ding H. MRMD2.0: A python tool for machine learning with feature ranking and reduction. *Current Bioinformatics*, 2020, **15**, 1213–1221.

[26] Yang S M, Deng B, Wang J, Li H Y, Lu M L, Che Y Q, Wei X L, Loparo K A. Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE transactions on neural networks and learning systems*, 2019, **31**, 148–162.

[27] Wang X F, Gao P, Liu Y F, Li H F, Lu F. Predicting thermophilic proteins by machine learning. *Current Bioinformatics*, 2020, **15**, 493–502.

[28] Sun Y, Yen G G, Yi Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2019, **23**, 173–187.

[29] Sun Y, Xue B, Zhang M, Yen G G, Lv J. Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 2020, 1–15.

[30] Bai B, Guo Z W, Zhou C, Zhang W, Zhang J Y. Application of adaptive reliability importance sampling-based extended domain PSO on single mode failure in reliability engineering. *Information Sciences*, 2021, **546**, 42–59.

[31] Yang B, Wang J B, Yu L, Shu H C, Yu T, Zhang X S, Yao W, Sun L M. A critical survey on proton exchange membrane fuel cell parameter estimation using meta-heuristic algorithms. *Journal of Cleaner Production*, 2020, **265**, 121660.

[32] Holland J H. Genetic algorithms. *Scientific American*, 1992, **267**, 66–72.

[33] Storn R, Price K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, **11**, 341–359.

[34] Zhao C H, Li J Y. Equilibrium selection under the bayes-based strategy updating rules. *Symmetry*, 2020, **12**, 739.

[35] Koza J R, Rice J P. Automatic programming of robots using genetic programming. *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI Press, San Jose, California, 1992, 194–201.

[36] Hansen N, Ller S D M, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 2003, **11**, 1–18.

[37] Xin Y, Yong L, Guangming L. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, **3**, 82–102.

[38] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing. *Readings in Computer Vision*, Fischler M A, Firschein O, eds., Morgan Kaufmann, San Francisco, USA, 1987, 606–615.

[39] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm. *Information Sciences*, 2009, **179**, 2232–2248.

[40] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, **69**, 46–61.

[41] Yang X S. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization* (*NISCO* 2010), 2010, **284**, 65–74.

[42] Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, 2013. **29**, 17–35.

[43] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 2007, **39**, 459–471.

[44] Li S M, Chen H L, Wang M J, Heidari A A, Mirjalili S. Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 2020, **111**, 300–323.

[45] Heidari A A, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H L. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems-the International Journal of Escience*, 2019, **97**, 849–872.

[46] Yang Y T, Chen H L, Asghar Heidari A, Gandomi A H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, 2021, https://doi.org/10.1016/j.eswa.2021.114864.

[47] Gandomi A H, Alavi A H. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 2012, **17**, 4831–4845.

[48] Wang G G. Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 2018, **10**, 151–164.

[49] Feng Y H, Deb S, Wang G G, Alavi A H. Monarch butterfly optimization: A comprehensive review. *Expert Systems with Applications*, 2021, **168**, 114418.

[50] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 2015, **89**, 228–249.

[51] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi A H. Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 2020, **152**, 113377.

[52] Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, **95**, 51–67.

[53] Glover F, Marti R. Tabu Search, in Metaheuristic Procedures for Training Neutral Networks, Alba E, Martí R, eds., Springer, Boston, USA, 2006, 53–69.

[54] Rao R V, Savsani V J, Vakharia D P, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 2011, **43**, 303–315.

[55] Gao N S, Luo D D, Cheng B Z, Hou H. Teaching-learning-based optimization of a composite metastructure in the 0–10 kHz broadband sound absorption range. *The Journal of the Acoustical Society of America*, 2020, **148**, EL125-EL129.

[56] Wolpert D H, Macready W G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, **1**, 67–82.

[57] Anholt B, Werner E. Interaction between food availability and predation mortality mediated by adaptive behavior. *Ecology*, 1995, **76**, 2230–2234.

[58] Cooper S. Optimal hunting group size: The need for lions to defend their kills against loss to spotted hyaenas. *African Journal of Ecology*, 1991, **29**, 130–136.

[59] Persons M H, Walker S E, Rypstra A L, Marshall S D. Wolf spider predator avoidance tactics and survival in the presence of diet-associated predator cues (Araneae: Lycosidae). *Animal Behaviour*, 2001, **61**, 43–51.

[60] Coleman S L, Brown V R, Levine D S, Mellgren R L. A neural network model of foraging decisions made under predation risk. *Cognitive*, *Affective*, & *Behavioral Neuroscience*, 2005, **5**, 434–451.

[61] Khater M, Murariu D, Gras R. Predation risk tradeoffs in prey: effects on energy and behaviour. *Theoretical Ecology*, 2016, **9**, 251–268.

[62] Shi K B, Wang J, Tang Y Y, Zhong S M. Reliable asynchronous sampled-data filtering of T–S fuzzy uncertain delayed neural networks with stochastic switched topologies. *Fuzzy Sets and Systems*, 2020, **381**, 1–25.

[63] Shi K B, Wang J, Zhong S M, Tang Y Y, Cheng J. Non-fragile memory filtering of TS fuzzy delayed neural networks based on switched fuzzy sampled-data control. *Fuzzy Sets and Systems*, 2020, **394**, 40–64.

[64] Shi K B, Tang Y Y, Zhong S M, Yin C, Huang X G, Wang W Q. Nonfragile asynchronous control for uncertain chaotic Lurie network systems with Bernoulli stochastic process. *International Journal of Robust and Nonlinear Control*, 2018, **28**, 1693–1714.

[65] Ni T, Chang H, Song T, Xu Q, Huang Z, Liang H, Yan A, Wen X. Non-intrusive online distributed pulse shrinking-based interconnect testing in 2.5D IC. *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, 2020, **67**, 2657–2661.

[66] Zhang H, Qiu Z, Cao J, Abdel-Aty M, Xiong L. Event-triggered ysnchronization for neutral-type semi-markovian neural networks with partial mode-dependent time-varying delays. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, **31**, 4437–4450.

[67] Hu J, Chen H L, Heidari A A, Wang M J, Zhang X Q, Chen Y, Pan Z F. Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowledge-Based Systems*, 2021, **213**, 106684.

[68] Shan W F, Qiao Z L, Heidari A A, Chen H L, Turabieh H, Teng Y T. Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. *Knowledge-Based Systems*, 2020, 106728.

[69] Tu J Z, Chen H L, Liu J C, Heidari A A, Zhang X Q, Wang M J, Ruby R, Pham Q-V. Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance. *Knowledge-Based Systems*, 2021, **212**, 106642.

[70] Zhao D, Liu L, Yu F H, Heidari A A, Wang M J, Liang G X, Muhammad K, Chen H. Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. *Knowledge-Based Systems*, 2020, https://doi.org/10.1016/j.knosys.2020.106510.

[71] Zhang Y N, Liu R J, Heidari A A, Wang X, Chen Y, Wang M J, Chen H L. Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing*, 2020.

[72] Chen H, Heidari A A, Chen H L, Wang M J, Pan Z F, Gandomi A H. Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. *Future Generation Computer Systems*, 2020, **111**, 175–198.

[73] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 2016, **96**, 120–133.

[74] Mirjalili S, Gandomi A H, Mirjalili S Z, Saremi S, Faris H, Mirjalili S M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 2017, **114**, 163–191.

[75] Kennedy J. *Particle Swarm Optimization*, *in Encyclopedia of Machine Learning*, Sammut C, Webb G I, eds., Springer, Boston, USA, 2010, 760–766.

[76] Yang X S. Firefly Algorithms for multimodal optimization. *Stochastic Algorithms*: *Foundations and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2009.

[77] Gupta S, Deep K. A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications*, 2019, **119**, 210–230.

[78] Abd Elaziz M, Oliva D, Xiong S. An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications*, 2017, **90**, 484–500.

[79] Sun Y, Wang X, Chen Y, Liu Z. A modified whale optimization algorithm for large-scale global optimization problems. *Expert Systems with Applications*, 2018, **114**, 563–577.

[80] Tubishat M, Abushariah M A M, Idris N, Aljarah I. Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Applied Intelligence*, 2018.

[81] Yong J, He F, Li H, Zhou W. A novel bat algorithm based on collaborative and dynamic learning of opposite population.

IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (*CSCWD*), 2018.

[82] Adarsh B R, Raghunathan T, Jayabarathi T, Yang X S, Economic dispatch using chaotic bat algorithm. *Energy*, 2016, **96**, 666–675.

[83] Jia D L, Zheng G X, Qu B Y, Khan M K. A hybrid particle swarm optimization algorithm for high-dimensional problems. *Computers & Industrial Engineering*, 2011, **61**, 1117–1122.

[84] Chen W, Zhang J, Lin Y, Chen N, Zhan Z, Chung H S, Li Y, Shi Y. Particle swarm optimization with an aging leader and challengers. *IEEE Transactions on Evolutionary Computation*, 2013, **17**, 241–258.

[85] Jia D L, Zheng G X, Khurram Khan M. An effective memetic differential evolution algorithm based on chaotic local search. *Information Sciences*, 2011, **181**, 3175–3187.

[86] Kaveh A, Khayatazad M. A new meta-heuristic method: Ray optimization. *Computers & Structures*, 2012, **112-113**, 283–294.

[87] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics & Computation*, 2007, **188**, 1567–1579.

[88] Huang F Z, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, 2007, **186**, 340–356.

[89] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 2007, **20**, 89–99.

[90] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation*, 2007, **186**, 1407–1422.

[91] Kannan B K, Kramer S N. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 1994, **116**, 405–411.

[92] Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, 1990, **112**, 223–229.

[93] Wang G G. Adaptive response surface method using inherited Latin hypercube design points. *Journal of Mechanical Design*, 2003, **125**, 210–220.

[94] Cheng M Y, Prayogo D. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures*, 2014, **139**, 98–112.

[95] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 2012, **110-111**, 151–166.

[96] Savsani P, Savsani V. Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling*, 2016, **40**, 3951–3978.

Appendix A

Table A1 Description of the 23 benchmark functions

Function	Function equation	Dim	Range	$f_{\min}$
F1	$f_1(x)=\sum_{i=1}^{n} x^2$	30	[−100,100]	0
F2	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[−10,10]	0
F3	$f_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
F4	$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	[−100,100]	0
F5	$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[−30,30]	0
F6	$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[−100,100]	0
F7	$f_7(x)=\sum_{i=1}^{n} ix^4+random[0,1)$	30	[−1.28,1.28]	0
F8	$f_8(x)=\sum_{i=1}^{n}-x\sin(\sqrt{\lvert x_i\rvert})$	30	[−500,500]	− 418.982
F9	$f_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[−5.12,5.12]	0
F10	$f_{10}(x)=-20e^{-0.2\sqrt{\frac{1}{\pi}\sum_{i=1}^{n}x_i}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}+20+e$	30	[−32,32]	0

F11	$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^{2}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[−600,600]	0
F12	$f_{12}(x)=\frac{\pi}{n}\{10\sin(ay_1)+\sum_{i=1}^{n=1}(y_i-1)^2[1+\sin^2(\pi y_{i+1})]+$ $(y_n-1)^2+\sum_{i=1}^{n}\mu(x_i,10,100,4)\}$	30	[−50,50]	0
F13	$f_{13}(x)=0.1\{\sin^2(3\pi x_i)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+$ $(x_n-1)^2[1+\sin^2(2\pi x_n)]+\sum_{i=1}^{n}\mu(x_i,5,100,4)\}$	30	[−50,50]	0
F14	$f_{14}(x)=(\frac{1}{5000}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i+a_{ij})^6})^{-1}$	2	[−65,65]	1
F15	$f_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^{2}+b_ix_2)}{b_i^{2}+b_ix_3+x_4}]^2$	4	[−5,5]	0.00030
F16	$f_{16}(x)=4x_1^{2}-2.1x_1^{4}+\frac{1}{3}x_1^{6}+x_1x_2-4x_2^{2}+4x_2^{4}$	2	[−5,5]	−1.0316
F17	$f_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^{2}+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi}\cos(x_1)+10)$	2	[−5,5]	0.398
F18	$f_{18}(x)=[1+(x_1+x_2+1)^2\times(19-14x_1+3x_1^{2}-14x_2+6x_1x_2+3x_2^{2})]\times$ $[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^{2}+48x_2-36x_1x_2+27x_2^{2})]$	2	[−2,2]	3
F19	$f_{19}(x)=-\sum_{i=1}^{4}c_ie^{-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2}$	3	[1,3]	−3.86
F20	$f_{20}(x)=-\sum_{i=1}^{4}c_ie^{-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2}$	6	[0,1]	−3.32
F21	$f_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	−10.1532
F22	$f_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	−10.4028
F23	$f_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	−10.5363

Table A2 Description of IEEE CEC2014 functions

ID	Function equation	Dim	Range	$f_{\min}$
	Unimodal functions			
F24	Rotated High Conditioned Elliptic Function	30	[−100,100]	100
F25	Rotated Bent Cigar Function	30	[−100,100]	200
F26	Rotated Discus Function	30	[−100,100]	300
	Simple multimodal functions			
F27	Shifted and Rotated Rosenbrock’s Function	30	[−100,100]	400
F28	Shifted and Rotated Ackley’s Function	30	[−100,100]	500
F29	Shifted and Rotated Weierstrass Function	30	[−100,100]	600
F30	Shifted and Rotated Griewank’s Function	30	[−100,100]	700
F31	Shifted Restringing’s Function	30	[−100,100]	800
F32	Shifted and Rotated Restringing’s Function	30	[−100,100]	900
F33	Shifted Schwefel’s Function	30	[−100,100]	1000
F34	Shifted and Rotated Schwefel’s Function	30	[−100,100]	1100
F35	Shifted and Rotated Katsuura Function	30	[−100,100]	1200
F36	Shifted and Rotated HappyCat Function	30	[−100,100]	1300
F37	Shifted and Rotated HGBat Function	30	[−100,100]	1400
F38	Shifted and Rotated Expanded Griewank’s plus Rosenbrock’s Function	30	[−100,100]	1500
F39	Shifted and Rotated Expanded Scoffer's F6 Function	30	[−100,100]	1600
	Hybrid functions			
F40	Hybrid Function 1 (N = 3)	30	[−100,100]	1700
F41	Hybrid Function 2 (N = 3)	30	[−100,100]	1800
F42	Hybrid Function 3 (N = 4)	30	[−100,100]	1900
F43	Hybrid Function 4 (N= 4)	30	[-100,100]	2000
F44	Hybrid Function 5 (N = 5)	30	[−100,100]	2100
F45	Hybrid Function 6 (N = 5)	30	[−100,100]	2200
	Composition functions			
F46	Composition Function 1 (N = 5)	30	[−100,100]	2300
F47	Composition Function 2 (N = 3)	30	[−100,100]	2400
F48	Composition Function 3 (N = 3)	30	[−100,100]	2500
F49	Composition Function 4 (N = 5)	30	[−100,100]	2600
F50	Composition Function 5 (N = 5)	30	[−100,100]	2700
F51	Composition Function 6 (N = 5)	30	[−100,100]	2800
F52	Composition Function 7 (N = 3)	30	[−100,100]	2900
F53	Composition Function 8 (N = 3)	30	[−100,100]	3000

Table A3 Parameter settings of counterparts

Algorithm	Parameter settings	Parametric note
SCA	$A = 2$	A is a constant
SSA	$c_1 \in [2,0]; c_2 \in [0,1]; c_3 \in [0,1]$	c_1 is an exponential decreasing variable; c_2 and c_3 are random numbers
GWO	$a = [2,0]$	a is a linear decreasing variable
MFO	$b = 1; t = [-1,1]; a \in [-1,-2]$	b is a constant; t is a random number; a is a linear decreasing variable;
WOA	$a_1 = [2,0]; a_2 = [-2,-1]; b = 1$	a_1 and a_2 are linear decreasing variables; b is a constant
ABC	$limit = 300$	*limit* is a constant
GSA	$Rpower = 1; ElitistCheck = 1; Rnorm = 2$	*Rpower ElitistCheck* and *Rnorm* are constants
FA	$a = 0.5; \beta = 0.2; \gamma = 1$	α and β are constants; γ means absorption coefficient
PSO	$c_1 = 2; c_2 = 2; vMax = 6$	c_1, c_2 and *vMax* are constants
DE	$beta_{\min} = 0.2; beta_{\max} = 0.8; pCR = 0.2$	*beta* means scaling factor; *pCR* means crossover probability
BA	$A = [1,2]; r = [0,1]; Q_{\min} = 0; Q_{\max} = 2$	A means loudness; r means pulse rate; Q means Frequency

Table A4 *p*-value of Wilcoxon test obtained from comparison with traditional algorithms on 53 functions

Function	PSO	DE	ABC	SCA	SSA	BA	FA	MFO
F1	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.697E−06
F2	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.601E−06
F3	1.734E-06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.691E−06
F4	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F5	1.734E−06	2.105E−03	1.734E−06	1.734E−06	4.114E−03	1.921E−06	1.734E−06	4.860E−05
F6	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**4.165E−01**
F7	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F8	1.734E−06	3.374E−04	9.766E−04	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F9	1.734E−06	1.250E−01	**1.000E+00**	**1.000E+00**	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F10	1.734E−06	1.962E−07	6.634E−07	1.730E−06	1.734E−06	1.734E−06	1.734E−06	1.673E−06
F11	1.734E−06	**1.000E+00**	2.189E−06	**1.000E+00**	1.734E−06	1.734E−06	1.734E−06	2.696E−05
F12	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.478E−05
F13	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	4.178E−04
F14	1.734E−06	**1.000E+00**	**1.000E+00**	1.734E−06	**1.000E+00**	1.734E−06	1.734E−06	4.883E−04
F15	1.734E−06	5.706E−02	1.734E−06	1.734E−06	2.879E−06	1.734E−06	1.734E−06	5.302E−05
F16	1.734E−06	**1.000E+00**	**1.000E+00**	1.734E−06	3.906E−03	1.734E−06	1.734E−06	**1.000E+00**
F17	1.734E−06	**1.000E+00**	**1.000E+00**	1.734E−06	**1.000E+00**	1.734E−06	1.734E−06	**1.000E+00**
F18	1.734E−06	3.906E−02	5.791E−04	1.734E−06	9.987E−07	1.734E−06	1.734E−06	**5.737E−02**
F19	1.734E−06	**1.000E+00**	**1.000E+00**	1.734E−06	6.250E−02	1.734E−06	1.734E−06	**1.000E+00**
F20	1.734E−06	**5.000E−01**	**1.000E+00**	1.734E−06	1.732E−06	1.734E−06	1.734E−06	6.648E−06
F21	7.731E−03	1.563E−02	1.563E−02	1.734E−06	**3.709E−01**	**1.414E−01**	**3.709E−01**	1.451E−04
F22	**8.936E−01**	4.883E−04	4.883E−04	4.286E−06	**2.210E−01**	**4.908E−01**	**2.452E−01**	**6.837E−01**
F23	**8.290E−01**	4.883E−04	4.883E−04	4.897E−04	**2.059E−01**	**6.583E−01**	**2.623E−01**	**4.403E−01**
F24	1.734E−06	1.734E−06	1.734E−06	1.734E−06	4.729E−06	9.711E−05	1.734E−06	1.734E−06
F25	1.734E−06	3.882E−06	3.515E−06	1.734E−06	**7.813E−01**	1.734E−06	1.734E−06	1.734E−06
F26	1.734E−06	3.854E−03	2.879E−06	1.734E−06	2.304E−02	3.882E−06	1.734E−06	1.734E−06
F27	2.623E−01	6.035E−03	2.255E−03	1.734E−06	6.564E−02	2.304E−02	1.734E−06	1.734E−06
F28	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	2.353E−06
F29	6.984E−06	**4.048E−01**	5.706E−04	1.734E−06	**6.435E−01**	1.734E−06	1.734E−06	3.182E−06
F30	1.734E−06	6.708E−06	4.416E−04	1.734E−06	**1.779E−01**	1.734E−06	1.734E−06	1.734E−06

F31	1.734E−06	2.232E−03	3.738E−05	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F32	1.734E−06	3.162E−03	2.603E−06	1.734E−06	**3.086E−01**	1.734E−06	1.734E−06	1.921E−06
F33	1.734E−06	1.734E−06	8.919E−05	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F34	1.734E−06	1.734E−06	4.072E−05	1.734E−06	3.405E−05	1.734E−06	1.734E−06	2.353E−06
F35	1.734E−06	1.734E−06	4.729E−06	1.734E−06	6.339E−06	1.734E−06	1.734E−06	1.734E−06
F36	3.065E−04	5.752E−06	1.734E−06	1.734E−06	**3.493E−01**	**6.583E−01**	1.734E−06	2.353E−06
F37	6.035E−03	**4.048E−01**	1.734E−06	1.734E−06	3.501E−02	**1.064E−01**	1.734E−06	1.734E−06
F38	7.712E−04	**3.389E−01**	1.287E−03	1.734E−06	2.957E−03	3.182E−06	1.734E−06	1.734E−06
F39	1.734E−06	1.734E−06	**2.536E−01**	1.734E−06	4.729E−06	1.734E−06	1.734E−06	1.921E−06
F40	1.734E−06	1.734E−06	1.734E−06	1.734E−06	2.067E−02	6.035E−03	1.734E−06	7.691E−06
F41	1.734E−06	2.585E−03	4.534E−04	1.734E−06	8.590E−02	1.734E−06	1.734E−06	1.359E−04
F42	2.843E−05	**1.714E−01**	6.320E−05	1.734E−06	3.405E−05	2.597E−05	1.734E−06	1.734E−06
F43	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F44	6.639E−04	1.734E−06	3.515E−06	1.734E−06	**1.714E−01**	**5.577E−01**	1.734E−06	3.882E−06
F45	1.251E−04	2.353E−06	1.127E−05	4.729E−06	**8.972E−02**	2.879E−06	3.882E−06	1.973E−05
F46	1.734E−06	4.320E−08	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F47	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F48	1.734E−06	1.734E−06	1.734E−06	2.563E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F49	3.182E−06	1.734E−06	2.585E−03	1.734E−06	1.397E−02	**6.435E−01**	1.734E−06	2.353E−06
F50	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F51	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F52	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.722E−06
F53	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
RANK	1	2	3	11	4	9	12	8

Table A4 (continued) *p*-value of Wilcoxon test obtained from comparison with traditional algorithms on 53 functions

Function	WOA	GWO	GSA
F1	**1.000E+00**	**1.000E+00**	1.734E−06
F2	**1.000E+00**	**1.000E+00**	1.734E−06
F3	1.734E−06	2.563E−06	1.734E−06
F4	1.734E−06	1.734E−06	1.734E−06
F5	1.734E−06	1.734E−06	1.734E−06
F6	1.734E−06	1.734E−06	1.734E−06
F7	1.359E−04	3.589E−04	1.734E−06
F8	1.734E−06	1.734E−06	1.734E−06
F9	**1.000E+00**	**1.000E+00**	1.734E−06
F10	6.104E−05	2.572E−07	1.734E−06
F11	**1.000E+00**	**1.000E+00**	1.734E−06
F12	1.734E−06	1.734E−06	1.734E−06
F13	1.734E−06	1.734E−06	1.734E−06
F14	3.196E−06	1.734E−06	1.734E−06
F15	1.734E−06	1.734E−06	1.734E−06
F16	2.441E−04	1.734E−06	1.734E−06
F17	1.734E−06	1.734E−06	1.734E−06
F18	1.734E−06	1.734E−06	1.734E−06
F19	1.734E−06	1.734E−06	1.734E−06
F20	1.734E−06	1.734E−06	1.734E−06
F21	**3.709E−01**	**3.709E−01**	7.514E−05

F22	**2.059E−01**	**2.059E−01**	**7.521E−02**
F23	**2.059E−01**	**2.059E−01**	1.566E−02
F24	1.734E−06	1.734E−06	1.734E−06
F25	1.734E−06	1.734E−06	1.734E−06
F26	1.734E−06	1.734E−06	1.734E−06
F27	1.734E−06	1.734E−06	**4.779E−01**
F28	1.734E−06	1.734E−06	1.734E−06
F29	1.734E−06	1.799E−05	1.734E−06
F30	1.734E−06	1.734E−06	1.734E−06
F31	1.734E−06	1.734E−06	1.734E−06
F32	1.921E−06	9.711E−05	1.734E−06
F33	1.734E−06	1.734E−06	1.734E−06
F34	1.734E−06	1.566E−02	5.216E−06
F35	1.734E−06	3.112E−05	1.734E−06
F36	**6.884E−01**	5.752E−06	1.734E−06
F37	4.196E−04	6.156E−04	**5.984E−02**
F38	1.734E−06	3.724E−05	4.492E−02
F39	1.734E−06	2.052E−04	1.734E−06
F40	1.734E−06	1.734E−06	1.734E−06
F41	7.712E−04	2.831E−04	1.734E−06
F42	1.494E−05	6.339E−06	6.156E−04
F43	1.734E−06	1.734E−06	1.734E−06
F44	1.734E−06	1.360E−05	8.188E−05
F45	1.477E−04	5.320E−03	1.734E−06
F46	3.790E−06	1.734E−06	1.734E−06
F47	1.734E−06	1.734E−06	1.734E−06
F48	1.318E−04	1.229E−05	1.734E−06
F49	2.849E−02	**8.972E−02**	5.307E−05
F50	1.734E−06	1.734E−06	1.734E−06
F51	2.563E−06	1.734E−06	1.734E−06
F52	1.720E−06	1.734E−06	1.734E−06
F53	1.734E−06	1.734E−06	1.734E−06
RANK	6	5	7

Table A5 *p*-value of Wilcoxon test obtained from comparison with advanced algorithms on 53 functions

Fun	m_SCA	OBSCA	MWOA	IWOA	CDLOB	CBA	CGPSO	ALCPSO	DECLS
F1	**1.000E+00**	1.734E−06	1.734E−06	**1.000E+00**	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F2	**1.000E+00**	1.734E−06	1.734E−06	**1.000E+00**	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F3	4.378E−04	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F4	1.734E−06	1.734E−06	2.931E−04	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F5	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**7.036E−01**	6.892E−05	1.734E−06	**7.499E−01**	1.734E−06
F6	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F7	**2.623E−01**	1.734E−06	1.734E−06	4.070E−02	1.734E−06	1.734E−06	5.287E−04	1.734E−06	6.836E−03
F8	1.734E−06	1.734E−06	1.713E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.730E−06	5.037E−04
F9	**1.000E+00**	**1.000E+00**	1.734E−06	**1.000E+00**	1.734E−06	1.734E−06	1.733E−06	1.734E−06	**2.500E−01**
F10	6.799E−08	4.320E−08	1.734E−06	6.334E−05	1.713E−06	1.734E−06	1.734E−06	1.709E−06	6.799E−08
F11	**1.000E+00**	**1.000E+00**	1.734E−06	**1.000E+00**	1.734E−06	5.606E−06	1.734E−06	2.696E−05	**1.000E+00**

F12	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**7.813E−01**	1.734E−06
F13	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**6.432E−01**	1.734E−06
F14	1.734E−06	1.734E−06	1.734E−06	5.388E−07	1.734E−06	8.277E−06	1.645E−06	**1.000E+00**	**5.000E−01**
F15	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	3.106E−05	**9.754E−01**
F16	1.734E−06	1.734E−06	1.734E−06	7.813E−03	1.734E−06	1.734E−06	1.734E−06	**1.000E+00**	**1.000E+00**
F17	1.734E−06	1.734E−06	1.734E−06	1.733E−06	1.734E−06	1.734E−06	1.734E−06	**1.000E+00**	**1.000E+00**
F18	1.734E−06	1.734E−06	1.734E−06	1.733E−06	1.734E−06	1.734E−06	1.734E−06	5.699E−05	**1.000E+00**
F19	1.734E−06	1.734E−06	1.734E−06	1.732E−06	1.734E−06	1.734E−06	1.734E−06	**1.000E+00**	**1.000E+00**
F20	1.734E−06	3.112E−05	1.734E−06	1.734E−06	1.734E−06	5.216E−06	1.734E−06	7.813E−03	**1.000E+00**
F21	4.716E−02	4.716E−02	1.734E−06	4.716E−02	2.703E−02	1.833E−03	4.716E−02	**1.600E−01**	1.221E−04
F22	**1.109E−01**	**1.020E−01**	1.734E−06	**1.020E−01**	**9.590E−01**	**2.802E−01**	**1.020E−01**	4.272E−04	2.441E−04
F23	**6.733E−01**	**6.435E−01**	3.182E−06	**6.435E−01**	**7.521E−02**	6.035E−03	**6.435E−01**	1.953E−03	9.766E−04
F24	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**5.999E−01**	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F25	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.566E−02	**6.268E−02**	1.734E−06	9.316E−06	6.339E−06
F26	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	5.216E−06	1.734E−06	**9.368E−02**	1.245E−02
F27	1.734E−06	1.734E−06	1.734E−06	2.879E−06	9.842E−03	2.843E−05	**1.204E−01**	4.729E−06	6.892E−05
F28	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	2.353E−06	1.734E−06	1.734E−06	1.734E−06
F29	1.494E−05	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	**6.143E−01**	2.879E−06
F30	1.734E−06	1.734E−06	1.734E−06	1.734E−06	5.287E−04	2.957E−03	1.734E−06	2.255E−03	1.712E−06
F31F	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	9.208E−05
F32	6.339E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	5.667E−03	**5.304E−01**
F33	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F34	3.405E−05	1.734E−06	1.734E−06	1.921E−06	1.734E−06	1.734E−06	1.734E−06	6.639E−04	1.734E−06
F35	1.734E−06	1.734E−06	1.734E−06	1.734E−06	3.589E−04	1.734E−06	1.734E−06	2.353E−06	1.734E−06
F36	2.765E−03	1.734E−06	1.734E−06	**6.583E−01**	**1.064E−01**	**1.470E−01**	6.639E−04	**7.655E−01**	1.734E−06
F37	2.603E−06	1.734E−06	1.734E−06	9.627E−04	4.492E−02	**1.589E−01**	4.390E−03	2.304E−02	2.957E−03
F38	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.921E−06	1.359E−04	**4.165E-01**	**4.405E−01**
F39	1.127E−05	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	2.353E−06	1.734E−06
F40	1.734E−06	1.734E−06	1.734E−06	1.734E−06	8.188E−05	2.353E−06	1.921E−06	7.691E−06	1.734E−06
F41	1.734E−06	1.734E−06	1.734E−06	**1.254E−01**	3.327E−02	2.067E−02	1.734E−06	1.957E−02	9.711E−05
F42	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.114E−03	**1.846E−01**
F43	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F44	3.182E−06	1.734E−06	1.734E−06	1.734E−06	7.271E−03	3.160E−02	1.799E−05	**8.774E−01**	1.734E−06
F45	9.271E−03	2.879E−06	1.734E−06	7.731E−03	2.353E−06	3.182E−06	1.397E−02	**3.820E−01**	3.182E−06
F46	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F47	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F48	1.734E−06	**1.250E−01**	1.734E−06	1.964E−04	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F49	4.449E−05	1.734E−06	1.734E−06	**2.712E−01**	**7.813E−01**	**5.999E−01**	1.734E−06	6.035E−03	1.734E−06
F50	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F51	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F52	1.734E−06	1.734E−06	1.734E−06	1.663E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06
F53	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06	1.734E−06