Shaoling Zhang, Yongquan Zhou* and Qifang Luo

Elite Opposition-Based Cognitive Behavior Optimization Algorithm for Global Optimization

DOI 10.1515/jisys-2017-0046

Received February 16, 2017; previously published online June 29, 2017.

Abstract: This paper presents an elite opposition-based cognitive behavior optimization algorithm (ECO). The traditional COA is divided into three stages: rough search, information exchange and share, and intelligent adjustment process. In this paper, we introduce the elite opposition-based learning in the third stage of COA, with a view to avoid the latter congestion as well as to enhance the convergence speed. ECOA is validated by 23 benchmark functions and three engineering design problems, and the experimental results have proven the superior performance of ECOA compared to other algorithms in the literature.

Keywords: Cognitive behavior optimization, elite opposition-based learning, benchmark functions, engineering design problems, swarm intelligence.

1 Introduction

There are many methods for solving optimization problems, but the whole can be classified into two categories. One is using traditional optimization methods, such as dynamic programming method, Newton method, and conjugate gradient method [29]. However, when solving large-scale combinatorial optimization problems, it will show great limitations, and as the scale of the problem continues to expand, the traditional optimization methods cannot satisfy the need of solving complex problems. Another is that modern optimization methods, such as metaheuristic algorithms [30], are based on intuitive or empirical construction algorithms. This algorithm has the advantages of fast convergence speed and high stability and makes up for the deficiency of the traditional optimization method in solving complex combination optimization problems.

The cognitive behavior optimization algorithm (COA) was proposed by Li et al. [16]. The COA is inspired by the artificial bee colony (ABC) algorithm [1, 13] based on a general framework constituted from bees searching for food sources. Meanwhile, combined with the human social cognitive behavior and differential evolution (DE) [23] to form a detailed model of cognitive behavior optimization, the COA is divided into three sections and two groups. The process includes rough search, information exchange and share, and intelligent adjustment. The first step, rough search, is similar to the ABC in that the scouts search for the food source. In this part, the Gaussian random walk method and Lévy flight are used to balance the exploration and exploitation. The second step, information exchange and share, is similar to the ABC in that the employed foragers dance in the dance area to share information. In this behavior, it uses the improved crossover and mutation operation of DE, and the select probability P_c is proposed [23]. The final step, intelligent adjustment, is similar to the observation of bees according to food source select food. The two groups of population are the cognitive

*Corresponding author: Yongquan Zhou, College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China; and Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China, e-mail: yongquanzhou@126.com

Shaoling Zhang: College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

Qifang Luo: College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China; and Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

population (Cpop) and the memory population (Mpop); each was one-second of the population. The Cpop randomly searches for the food source in space, and the Mpop stores the information of the good food source found by the Cpop. The two groups work together to find the optimal solution.

In this paper, an elite opposition-based COA (ECO) is proposed. Twenty-three benchmark test functions and three popular structure engineering design problems are compared to illustrate the improved ECOA convergence effect and the optimization effects are obviously enhanced.

Section 2 briefly introduces the COA. In Section 3, the ECOA is presented. The simulation experiment and the test results are shown in Section 4. The conclusions and future work are described in Section 5. Finally, the acknowledgments are shown in Section 6.

2 COA

The COA was proposed by Li et al. [16]. Based on the behavior model of ABC and DE, the cognitive behavior model is proposed and it contains three main behaviors: rough search, information exchange and share, and intelligent adjustment. The COA is also divided into two groups: Cpop and Mpop; each was one-second of the population.

2.1 Rough Search

Before the step of rough search, according to the cognitive behavior model, the Cpop and Mpop are initialized as formulas (1) and (2):

$$Cpop_i = \text{rand} \cdot (\text{up} - \text{low}) + \text{low} \quad (1)$$

$$Mpop_i = \text{rand} \cdot (\text{up} - \text{low}) + \text{low} \quad (2)$$

where the $Cpop_i$ and $Mpop_i$ are the i th-generation Cpop and Mpop ($i=1, 2, 3, \dots, N$), respectively, N is the population size, up and low are the upper and lower boundaries of the search space, and rand is the random number of $[0, 1]$.

In this part, the Cpop searches for food sources in space using the Gaussian random walk or Lévy flight to generate the new individual around the current individual. The Gaussian random walk is used to expand the search space that can be enlarged, whereas the Lévy flight is used for faster convergence. At this stage, mining and detection are made into balance by mutual cooperation. The formulas are as follows:

$$Cpop'_i = \text{Gaussian}(G_{\text{best}}, \sigma) + (r_1 \cdot G_{\text{best}} - r_2 \cdot Cpop_i), \text{rand} \leq 0.5 \quad (3)$$

$$Cpop'_i = Cpop_i + \alpha \otimes \text{lévy}(s) \otimes (Cpop_i - G_{\text{best}}) | s = \frac{\mu}{|\nu|^{1/\beta}}, \text{rand} > 0.5 \quad (4)$$

$$\sigma = \left(\frac{\log(g)}{g} \right) \cdot (Cpop_i - G_{\text{best}}) \quad (5)$$

$$\sigma_\mu = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right)^{1/\beta}, \sigma_\nu = 1 \quad (6)$$

Here, G_{best} is the current best solution, the step size σ is calculated by $Cpop_i - G_{\text{best}}$, and $\log(g)/g$ is used to control the range of σ . r_1 and r_2 are random numbers on $[0, 1]$. In Eq. (4), α is the step-size scaling factor, where $\alpha = 0.01$; μ , ν is the parameter from the standard normal distribution, specifically Eq. (6), where $\beta = 3/2$.

2.2 Information Exchange and Share

In this stage, the $Mpop$ is used to store the original food information. This process draws on the crossover and mutation operations in DE [23]. The crossover probability P_c is used to determine how the $Cpop$ is updated, and then the good locations are stored in the $Mpop$. The specific steps are as follows.

Step 1: Calculate the crossover probability (P_{c_i}) according to Eq. (7):

$$P_{c_i} = \frac{\text{rank}(\text{fitCpop}_i)}{N/2} \quad (7)$$

Here, fitCpop_i is the fitness value of the $Cpop$ ($Cpop_i$). The $\text{rank}(\text{fitCpop}_i)$ is the ranking of the fitness value of the individual of the $Cpop$ ($Cpop_i$) according to the order from high to low.

Step 2: Update the $Mpop$:

$$\text{if } r_1 < r_2, \text{ then } Mpop = Cpop \quad (8)$$

$$Mpop = \text{permuting}(Mpop) \quad (9)$$

Step 3: The crossover probability (P_{c_i}) was used to select the update mode of the $Cpop$ ($Cpop_i$).

$$\begin{cases} Cpop_{i,j}'' = Cpop_{k,j} + \text{rand} \cdot (G_{\text{best},j} - Cpop_{i,j} + Mpop_{i,j} - Cpop_{h,j}), & \text{rand} \leq P_{c_i} \\ Cpop_{i,j}'' = Cpop_{i,j} + \text{rand} \cdot (Mpop_{i,j} - Cpop_{k,j}), & \text{rand} > P_{c_i} \end{cases} \quad (10)$$

Here, $i, k, h \in \{1, 2, 3, \dots, N/2\}$, $i \neq k \neq h$, $j \in \{1, 2, 3, \dots, D\}$, among them, D is the dimension, where Eq. (7) ranks the $Cpop$ ($Cpop_i$) by the fitness value from high to low, so the probability that the higher the crossover probability (P_{c_i}) is preserved. Eqs. (8) and (9) are crossover operations, and the original $Mpop$ is randomly ordered in formula (9) to ensure that the memory capacity of the population is constant. Eq. (10) borrows from the classic DE algorithm [23] in the classical mutation operation DE/rand/1, DE/best/1, and the two variants are used to ensure that the $Cpop$ can find a better solution location.

2.3 Intelligent Adjustment

This stage is to improve the ability to find the optimal solution to the $Cpop$ of individuals to update. After the first two stages, exploitation and exploration in this phase of the basic COA are balanced. The position of the individuals was adjusted by the information exchange among individuals. Here, it is emphasized that Eqs. (11) and (12) are updated under the condition of $\text{rand} > P_{c_i}$. If $\text{rand} \leq P_{c_i}$ is not updated, $Cpop$ still retains the original value.

$$Cpop_i''' = Cpop_i + \varphi \cdot (Cpop_i - G_{\text{best}}) \quad \text{rand} < 0.5 \quad (11)$$

$$Cpop_i''' = Cpop_i + \varphi \cdot (Cpop_i - Cpop_j) \quad \text{rand} > 0.5 \quad (12)$$

In the above formula, φ is the random number of $[-1, 1]$, and the adjustment of the $Cpop$ through Eqs. (11) and (12) makes it easier to find the global most position.

Based on the above three stages, the whole model of the $Cpop$ optimization algorithm is constructed, and its pseudo-code is shown in Algorithm 1, where N is the population number. The number of the $Cpop$ and $Mpop$ is $N/2$, and D is the dimension.

Algorithm 1: Pseudo-code of the COA.**Start**

```

Initialize the populations Cpop and Mpop according to Eqs. (1) and (2);
g = 1;
//Rough Search
for i from 1 to N/2 do
    if rand ≤ 0.5, then
        Cpop'_i = Gaussian(G_best, σ) + (r_1 · G_best - r_2 · Cpop_i) //r_1 and r_2 are random values;
    else
        Cpop'_i = Cpop_i + α ⊗ lévy(s) ⊗ (Cpop_i - G_best) | s =  $\frac{\mu}{|\nu|^{1/\beta}}$ ;
    end-if
end-for
//Information Exchange and Share
Calculate the P_c through P_{c_i} = (rank(fitCpop)) / (N/2);
if r_1 < r_2, then Mpop = Cpop, end-if
Mpop = permuting(Mpop)
for i from 1 to N/2 do
    for j from 1 to D do
        if rand ≤ P_{c_i}, then
            Cpop''_{i,j} = Cpop_{k,j} + rand · (G_{best,j} - Cpop_{i,j} - Mpop_{i,j} - Cpop_{h,j});
        else
            Cpop''_{i,j} = Cpop_{i,j} + rand · (Mpop_{i,j} - Cpop_{k,j});
        end-if
    end-for
end-for
//Intelligent Adjustment
Calculate the P_c through P_{c_i} = (rank(fitCpop)) / (N/2);
for i from 1 to N/2 do
    if rand > P_{c_i}, then
        if rand < 0.5, then
            Cpop'''_i = Cpop_i + φ · (Cpop_i - G_best);
        else
            Cpop'''_i = Cpop_i + φ · (Cpop_i - Cpop_j);
        end-if
    end-if
end-for
Memorize the best solution achieved so far and exist main repeat;
g = g + 1;
until Fes = MaxFes

```

End

3 ECOA

The standard COA (as proposed by Li et al. [16]) is outstanding in solving low-dimensional multimodal problems but is prone to postcongestion in high-dimensional problems and has been trapped in local optimal solutions. Therefore, this chapter introduces the elite opposite strategy in the intelligent adjustment phase

of the basic COA, reduces the late congestion degree through the elite opposite strategy, and speeds up the convergence speed. This algorithm is called the ECOA.

3.1 Opposition-Based Learning (OBL)

In 2005, Professor Tizhoosh proposed the concept of OBL [24]. The main idea of OBL was to consider each individual candidate while considering its opposite individual, which might be closer to the optimal individual. The OBL strategy can effectively improve the population diversity and avoid the premature convergence of the algorithm. General intelligent algorithms randomly generate initial population, then approach to the optimal solution gradually, and eventually find or close to the optimal solution. In the search process, while searching for the current solution and opposite solution, choose a better solution for the next generation of groups that greatly improve the efficiency of the algorithm.

Opposition solution: Suppose that there is a real number x on the range of $[a, b]$, then the opposite of x is defined as $x' = a + b - x$. Based on this, it is assumed that there is a solution point $p = (x_1, x_2, \dots, x_d)$ of a D dimension on the region R , $x_j \in [a_j, b_j]$, and then define $p' = (x'_1, x'_2, \dots, x'_d)$ for the opposite solution of p . Among them, $x'_j = k * (a_j + b_j) - x_j$, k is the random number of $[0, 1]$, also known as the generalization coefficient.

3.2 Elite OBL Strategies

The OBL can better expand the search range of the population and improve the performance of the algorithm. However, the OBL has a certain randomness in generating its opposite solution. Each randomly generated candidate compared to its opposite individual has a probability of 50% away from the optimal solution of the problem. The current optimal individual in the cognitive group is regarded as the elite individual, and the opposite solution is generated by the elite OBL. The sharing of the information among the individuals can make the cognitive individual better search the global optimal value.

The reverse learning mechanism of elite is a new search strategy in the field of evolutionary computation. Its guiding ideology is to evaluate a feasible solution and simultaneously evaluate the solution of reverse mapping. From these two alternative solutions, a better solution is chosen as the next generation of the solution. In this chapter, the solution with the best fitness value in the population is defined as the elite cognitive individual, which is expressed as $Cpop_e = (Cpop_{e,1}, Cpop_{e,2}, \dots, Cpop_{e,D})$.

A cognitive individual in the population ($Cpop_i$) and a cognitive individual obtained by inverse mapping ($Cpop'_i$) can be expressed as $Cpop_i = (Cpop_{i,1}, Cpop_{i,2}, \dots, Cpop_{i,D})$ and $Cpop'_i = (Cpop'_{i,1}, Cpop'_{i,2}, \dots, Cpop'_{i,D})$, respectively. $Cpop_e$, $Cpop_i$, and $(Cpop'_i)$ satisfy Eq. (13). Among them, n is the population size, D is the search space dimension, $k \in U(0, 1)$, and db_j and da_j are the upper and lower bounds of the first decision variable, which can be calculated by Eq. (14):

$$Cpop'_{i,j} = k * (da_j + db_j) - Cpop_{e,j}, i = 1, 2, \dots, n/2, j = 1, 2, \dots, D \quad (13)$$

$$da_j = lb, db_j = ub \quad (14)$$

In this case, we need to use the boundary control strategy to prevent the perceptual individuals of the reverse mapping ($Cpop'_i$) from jumping out (da_j, db_j). Therefore, we can adjust it using Eq. (15):

$$Cpop'_{i,j} = \text{rand}(da_j, db_j) \text{ if } Cpop'_{i,j} < da_j \text{ or } db_j > Cpop'_{i,j} \quad (15)$$

Specific implementation steps of the elite opposition-based flower pollination algorithm (EOFPA) can be summarized in the pseudo-code shown in Algorithm 2.

Algorithm 2: Pseudo-code of the ECOA.**Start**

Initialize the populations Cpop and Mpop according to Eqs. (1) and (2);

$g = 1$;

//Rough Search

for i from 1 to $N/2$ **do**

if $\text{rand} \leq 0.5$, **then**

$\text{Cpop}'_i = \text{Gaussian}(G_{\text{best}}, \sigma) + (r_1 \cdot G_{\text{best}} - r_2 \cdot \text{Cpop}_i)$ // r_1 and r_2 are random values;

else

$\text{Cpop}'_i = \text{Cpop}_i + \alpha \otimes \text{lévy}(s) \otimes (\text{Cpop}_i - G_{\text{best}}) \mid s = \frac{\mu}{|\nu|^{1/\beta}}$;

end-if

end-for

//Information Exchange and Share

Calculate the P_c through $P_{c_i} = (\text{rank}(\text{fitCpop}_i)) / (N/2)$;

if $r_1 < r_2$, **then** $\text{Mpop} = \text{Cpop}$, **end-if**

$\text{Mpop} = \text{permuting}(\text{Mpop})$

for i from 1 to $N/2$ **do**

for j from 1 to D **do**

if $\text{rand} \leq P_{c_i}$ **then**

$\text{Cpop}''_{i,j} = \text{Cpop}_{k,j} + \text{rand} \cdot (G_{\text{best},j} - \text{Cpop}_{i,j} + \text{Mpop}_{i,j} - \text{Cpop}_{h,j})$;

else

$\text{Cpop}''_{i,j} = \text{Cpop}_{i,j} + \text{rand} \cdot (\text{Mpop}_{i,j} - \text{Cpop}_{k,j})$;

end-if

end-for

end-for

//Intelligent Adjustment

Calculate the P_c through $P_{c_i} = (\text{rank}(\text{fitCpop}_i)) / (N/2)$;

for i from 1 to $N/2$ **do**

if $\text{rand} > P_{c_i}$, **then**

if $\text{rand} < 0.5$, **then**

$\text{Cpop}'''_i = \text{Cpop}_i + \varphi \cdot (\text{Cpop}_i - G_{\text{best}})$;

else

$\text{Cpop}'''_i = \text{Cpop}_i + \varphi \cdot (\text{Cpop}_i - \text{Cpop}_j)$;

end-if

$\text{Cpop}'_{i,j} = k \cdot (da_j + db_j) - \text{Cpop}_{e,j}$, $i = 1, 2, \dots, n/2$, $j = 1, 2, \dots, D$

if $\text{Cpop}'_{i,j} < da_j$ or $db_j > \text{Cpop}'_{i,j}$

$\text{Cpop}'_{i,j} = \text{rand}(da_j, db_j)$

end-if

end-if

end-for

 Memorize the best solution achieved so far and exist main repeat;

$g = g + 1$;

until $FES = \text{MaxFES}$

End

4 Simulation Experiments and Result Analysis

To verify the improved ECOA from many aspects, 23 classical test functions and three engineering examples were selected to be compared.

All the experiments in this section were operated on computer with 3.30 GHz Intel® Core™ i5-4590 processor and 4 GB of RAM using MATLAB R2012a.

4.1 Functions Test

4.1.1 Benchmark Test Functions

To verify the effectiveness of the algorithm, 23 standard test functions [6, 15, 17] were tested to reflect the objectivity of the experimental results.

These 23 standard test functions can be divided into three types (high-dimensional unimodal function, high-dimensional multimodal function, and low-dimensional function), where f_{01} to f_{07} are high-dimensional unimodal functions, f_{08} to f_{12} are high-dimensional multimodal functions, and f_{13} to f_{23} are low-dimensional functions. f_{05} is a classical test function whose global minimum value is at the bottom of a parabola and the fitness value of the position near the bottom of the trough changes little, so it is difficult to find the global minimum of the test function. f_{07} is a multidimensional multimodal function with a large number of local minima in its domain, which increases the difficulty of searching the objective function. In the 23 standard test functions selected in this chapter, the optimal value of most functions is zero, which can fully verify the optimization ability of the algorithm and also select the standard test function with some nonzero optimal values. The test functions and related configurations are shown in Table 1.

4.1.2 Results of Test Results Analysis

In the test, the number of population N is set to 50, the maximum number of iterations is 1000, independently run 30 times. In the test results table, Best, Mean, Worst, and Std, respectively, in 30 independent experiments, are the optimal solution, mean, worst-case solution, and standard deviation of the 30 run results. In addition, the dimension of the test function is described in the table. The rank of each test function is sorted by the variance value. Sum1 is the number of the first rankings in the statistics, and the last rank is the average of the variance. It is worth noting in the table the boldfaced underlined values for the test function of the optimal value.

Table 2 shows the results of the high-dimensional unimodal functions test. Tables 3 and 4 show the results of high-dimensional multimodal functions and low-dimensional functions, respectively. To verify the superiority of the function test results, the algorithm is compared to ABC [18], Cuckoo search (CS) [27], FPA [28], and Grey wolf optimizer (GWO) [19]. The following are the parameter settings of the comparison algorithms:

1. ABC: limit = 5D;
2. CS: $\beta = 1.5$, $\rho_0 = 1.5$;
3. FPA: $\rho = 0.8$;
4. GWO: Linearly α decreased from 2 to 0.

In Table 2, in the high-dimensional single-peak test functions, f_{01} to f_{04} found the optimal value, and its mean and Std are also 0; whereas the other comparison algorithms cannot test the five benchmark functions, it still found the optimal solution in a 30-dimensional space. f_{05} ranked third, while it is best to find that the results are smaller than the rest of the algorithm. In the f_{06} function, obviously, GWO find the results better than the rest of the algorithm. The f_{07} test results can be found, although the standard test results are not achieved, but the variance of the ECOA is optimal, and its optimal value is also obvious.

Figures 1–14 show the convergence and ANOVA test plots of the ECOA and other comparison algorithms in the f_{01} to f_{07} test functions. The logarithm of the Y-axis is taken in the convergence plot, and it is easy to find that the convergence rate is faster in f_{01} to f_{04} , f_{06} than in other algorithms. Although f_{05} convergence accuracy is good for GWO, the ECOA converges earlier than GWO. In the ANOVA test plots, although f_{05} is not as good as GWO and the COA, f_{01} to f_{04} , f_{06} are very stable compared to other contrasting algorithms.

In Table 3, f_{08} to f_{11} found the theoretical optimal solution, and the variance is 0 in f_{09} to f_{11} , which shows its high stability. Although f_{08} found the optimal value, its relatively large base caused by the corresponding

Table 1: Benchmark Test Functions.

Benchmark test functions	D	Range	Optimum
$f_{01}(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$f_{02}(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$f_{03}(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
$f_{04}(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	30	$[-100, 100]$	0
$f_{05}(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$f_{06}(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100, 100]$	0
$f_{07}(x) = \sum_{i=1}^n x_i^4 + \text{random}(0, 1)$	30	$[-1.28, 1.28]$	0
$f_{08}(x) = -\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-12569.5
$f_{09}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) \right) + 20 + e$	30	$[-32, 32]$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	$[-600, 600]$	0
$f_{12}(x) = \frac{\pi}{D} \left\{ \begin{aligned} &10 \sin^2(\pi y_1) \\ &+ \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i)] (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i)] \\ &+ (y_D - 1)^2 \end{aligned} \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - z)^m, & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$f_{13}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.0003075
$f_{14}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316285
$f_{15}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	2	$[-5.12, 5.12]$	-1
$f_{16}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-5, 5]$	3
$f_{17}(x) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	3	$[0, 1]$	-3.8628
$f_{18}(x) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$	6	$[0, 1]$	-3.32

Table 1 (continued)

Benchmark test functions	D	Range	Optimum
$f_{19}(x) = -\sum_{i=1}^5 [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$f_{20}(x) = -\sum_{i=1}^7 [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4029
$f_{21}(x) = -\sum_{i=1}^{10} [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5364
$f_{22}(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$	2	[-100, 100]	-1
$f_{23}(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	[-100, 100]	-1

Table 2: Simulation Results for Test Functions f_i , $i=1, 2, 3, 4, 5, 6, 7$.

Benchmark functions	Result	Method					
		ABC	CS	FPA	GWO	COA	ECOA
$f_{01}(D=30)$	Best	0.000572	0.003582902	0.387177862	8.4294E-150	1.076E-142	<u>0</u>
	Worst	0.013345	0.015870368	1.148099251	3.7913E-140	9.8669E-104	<u>0</u>
	Mean	0.005	0.009165081	0.728460406	1.2743E-141	5.7426E-105	<u>0</u>
	Std	0.003727	0.002955699	0.191680649	6.9199E-141	2.2095E-104	<u>0</u>
	rank	5	4	6	2	3	1
$f_{02}(D=30)$	Best	0.006895	0.642996775	1.517056239	1.43855E-83	4.79188E-73	<u>0</u>
	Worst	0.026708	2.34579469	3.03894779	4.34386E-80	2.03567E-48	<u>0</u>
	Mean	0.014761	1.314601235	2.277601766	4.45919E-81	6.79029E-50	<u>0</u>
	Std	0.004804	0.473421858	0.383790602	9.53865E-81	3.71652E-49	<u>0</u>
	rank	4	6	5	2	3	1
$f_{03}(D=30)$	Best	14078.44	268.9629877	2.272699762	3.15374E-77	5.2439E-139	<u>0</u>
	Worst	37939.76	626.9926428	8.879890954	3.91528E-62	8.54963E-95	<u>0</u>
	Mean	28852.36	453.5583574	4.882453968	1.30531E-63	2.84991E-96	<u>0</u>
	Std	4943.01	94.07702252	1.57114457	7.14825E-63	1.56094E-95	<u>0</u>
	rank	5	6	4	3	2	1
$f_{04}(D=30)$	Best	67.49763	1.629247498	0.687744025	1.29688E-48	5.43846E-69	<u>0</u>
	Worst	88.10366	6.121125683	1.17372478	2.00775E-44	9.76111E-47	<u>0</u>
	Mean	79.49636	3.071629218	0.962041686	1.35812E-45	3.33062E-48	<u>0</u>
	Std	4.782838	0.844484327	0.116680981	3.76964E-45	1.78093E-47	<u>0</u>
	rank	6	5	4	3	2	1
$f_{05}(D=30)$	Best	63.39128	30.02363766	143.3386217	4.944869599	19.57969669	1.20E-06
	Worst	259.3017	68.50521836	396.1431604	7.214109084	22.46671049	21.87445495
	Mean	134.4276	39.50931063	241.746718	5.993528429	21.01145488	15.24227714
	Std	53.42352	10.42249264	66.89327227	0.752141321	0.781294922	9.379551237
	rank	5	4	6	1	2	3
$f_{06}(D=30)$	Best	0.000178	0.003614324	0	0	2.41782E-22	3.54003E-25
	Worst	0.010919	0.019058352	3	0	4.85147E-17	2.04178E-21
	Mean	0.003574	0.008784236	0.9	0	1.77611E-18	2.80758E-22
	Std	0.002612	0.003671227	0.88473647	0	8.83302E-18	5.06707E-22
	rank	4	5	6	1	3	2
$f_{07}(D=30)$	Best	0.124748	0.01212532	0.895489753	1.70094E-05	0.000112593	6.67353E-07
	Worst	0.50385	0.066882382	7.677527996	0.000613755	0.003110998	0.000273177
	Mean	0.319918	0.035834524	3.315656718	0.000205695	0.001062594	8.16213E-05
	Std	0.091512	0.012337982	1.354025355	0.000169708	0.000791312	7.51741E-05
	rank	5	4	6	2	3	1

Table 3: Simulation Results for Test Functions f_i , $i = 8, 9, 10, 11, 12$.

Benchmark functions	Result	Method					
		ABC	CS	FPA	GWO	COA	ECOA
f_{08} ($D = 30$)	Best	-12928.3	-9236.781	-59.26595676	-3439.67801	-10713.9255	-12569.4866
	Worst	-10882.8	-8361.57903	-59.26593773	-2427.32615	-8423.38809	-9272.76399
	Mean	-11892.5	-8673.80683	-59.26595555	-2905.52286	-9668.18692	-12415.5099
	Std	427.5466	189.6242516	3.66702E-06	246.0091954	605.6102804	621.246281
	rank	4	2	1	3	5	6
f_{09} ($D = 30$)	Best	11.09179	59.70634524	4.303034334	0	0	0
	Worst	41.73442	109.3914666	91.26200005	3.093527585	0	0
	Mean	24.81539	85.96988736	33.5159609	0.103117586	0	0
	Std	6.649301	10.4501637	18.53734058	0.56479828	0	0
	rank	4	5	6	3	1	1
f_{10} ($D = 30$)	Best	6.487884	2.12739898	0.723999208	4.44089E-15	8.88178E-16	8.88178E-16
	Worst	12.2938	7.192223427	2.212411203	7.99361E-15	8.88178E-16	8.88178E-16
	Mean	9.795846	4.080766601	1.418156441	4.91459E-15	8.88178E-16	8.88178E-16
	Std	1.575547	1.310772991	0.364313386	1.22834E-15	0	0
	rank	6	5	4	3	1	1
f_{11} ($D = 30$)	Best	0.004203	0.031663188	0.009906228	0	0	0
	Worst	0.139289	0.241601554	0.035828502	0.061742336	0	0
	Mean	0.049833	0.088445467	0.023014852	0.011809657	0	0
	Std	0.036116	0.043991116	0.007264224	0.019333812	0	0
	rank	5	6	3	4	1	1
f_{12} ($D = 30$)	Best	6.79E-06	0.44067162	0.001559894	2.99552E-08	5.03621E-24	5.55916E-30
	Worst	0.001092	2.046704678	0.006489177	0.020127238	1.79269E-20	4.55893E-22
	Mean	0.000134	1.070493224	0.003429015	0.001873055	2.11792E-21	2.96744E-23
	Std	0.000206	0.377032703	0.001335403	0.005742838	3.88986E-21	8.55077E-23
	rank	4	6	3	5	2	1

Table 4: Simulation Results for Test Functions f_i , $i = 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23$.

Benchmark functions	Result	Method					
		ABC	CS	FPA	GWO	COA	ECOA
f_{13} ($D = 4$)	Best	0.000795	0.000307486	0.000308034	0.000307487	0.000307486	0.000307486
	Worst	0.001395	0.00030753	0.000320836	0.02036334	0.000307486	0.000424294
	Mean	0.001013	0.000307488	0.000310906	0.004389054	0.000307486	0.00031138
	Std	0.000162	8.16255E-09	3.43522E-06	0.008126933	2.08502E-19	2.13261E-05
	rank	5	2	3	6	1	4
f_{14} ($D = 2$)	Best	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-1.031628453
	Worst	-1.031628453	-1.031628453	-1.031628453	-1.031628441	-1.031628453	-1.031628453
	Mean	-1.031628453	-1.031628453	-1.031628453	-1.031628451	-1.031628453	-1.031628453
	Std	4.46E-16	6.77522E-16	6.51945E-16	2.90834E-09	6.77522E-16	6.77522E-16
	rank	1	3	2	6	3	3
f_{15} ($D = 2$)	Best	-1	-1	-1	-1	-1	-1
	Worst	-0.99992	-1	-1	-1	-1	-1
	Mean	-0.99999	-1	-1	-1	-1	-1
	Std	2.01E-05	1.08403E-12	0	0	0	0
	rank	6	5	1	1	1	1
f_{16} ($D = 2$)	Best	3	3	3	3	3	3
	Worst	3	3	3	3	3	3
	Mean	3	3	3	3	3	3
	Std	0.001135	1.87325E-15	1.34749E-15	3.57516E-06	1.28021E-15	1.96537E-15
	rank	6	3	2	5	1	4

Table 4 (continued)

Benchmark functions	Result	Method					
		ABC	CS	FPA	GWO	COA	ECO
$f_{17} (D=3)$	Best	-3.862782148	-3.862782148	-3.862782148	-3.86278206	-3.862782148	-3.862782148
	Worst	-3.862781981	-3.862782148	-3.862782148	-3.85489964	-3.862782148	-3.862782148
	Mean	-3.862782142	-3.862782148	-3.862782148	-3.86116516	-3.862782148	-3.862782148
	Std	1.49E-07	2.71009E-15	2.71009E-15	0.002856252	2.71009E-15	2.71009E-15
	rank	5	1	1	6	1	1
$f_{18} (D=6)$	Best	-3.321995172	-3.321995172	-3.32199517	-3.32199439	-3.32199517	-3.32199517
	Worst	-3.321995171	-3.321995172	-3.20310205	-3.08064935	-3.20310205	-3.20310205
	Mean	-3.321995172	-3.321995172	-3.26888982	-3.26221742	-3.30217965	-3.31010586
	Std	3.86E-11	1.09359E-13	0.058606391	0.075197206	0.045066321	0.036277689
	rank	2	1	5	6	4	3
$f_{19} (D=4)$	Best	-10.15319968	-10.15319968	-5.055197729	-10.1531021	-10.15319968	-10.15319968
	Worst	-10.0668242	-10.15319968	-5.055197729	-5.05518379	-10.15319968	-10.15319968
	Mean	-10.1467941	-10.15319968	-5.055197729	-8.96947353	-10.15319968	-10.15319968
	Std	0.018476	2.45741E-14	9.03362E-16	2.181833329	7.12072E-15	7.2269E-15
	rank	5	4	1	6	2	3
$f_{20} (D=4)$	Best	-10.4029406	-10.40294057	-5.087671825	-10.4028899	-10.40294057	-10.40294057
	Worst	-10.290525	-10.40294057	-5.087671825	-5.08766834	-10.40294057	-10.40294057
	Mean	-10.3958391	-10.40294057	-5.087671825	-10.0482985	-10.40294057	-10.40294057
	Std	0.006288	3.00029E-13	3.64717E-15	1.348448694	1.47518E-15	1.51161E-15
	rank	5	4	3	6	1	2
$f_{21} (D=4)$	Best	-10.5364098	-10.53640982	-5.128480787	-10.5363233	-10.53640982	-10.53640982
	Worst	-10.3324712	-10.53640982	-5.128480787	-5.12847474	-10.53640982	-10.53640982
	Mean	-10.525536	-10.53640982	-5.128480787	-9.99689365	-10.53640982	-10.53640982
	Std	0.005168	1.85544E-12	3.59458E-15	1.645250848	1.80672E-15	1.80672E-15
	rank	5	4	3	6	1	1
$f_{22} (D=2)$	Best	-1	-1	-0.01277964	-0.999999998	-1	-1
	Worst	-1	-1	-0.01277964	-0.999999721	-1	-1
	Mean	-1	-1	-0.01277964	-0.999999895	-1	-1
	Std	6.31E-10	0	0	7.66376E-08	0	0
	rank	5	1	1	6	1	1
$f_{23} (D=2)$	Best	-0.99995	-1	-1	-1	-1	-1
	Worst	-0.99028	-0.999971724	-1	-0.99028409	-1	-1
	Mean	-0.99696	-0.999997733	-1	-0.99805682	-1	-1
	Std	0.003355	5.65366E-06	0	0.003952802	0	0
	rank	5	4	1	6	1	1
	Sum1	1	3	6	3	11	14
	ave	4.5	3.875	3.4583	3.9583	2.3333	2.1467
RANK		6	4	3	5	2	1

Figure 1: $D=30$, Evolution Curves of Fitness Value for f_{01} .

Figure 2: $D=30$, ANOVA Test of Global Minimum for f_{01} .

Figure 3: $D=30$, Evolution Curves of Fitness Value for f_{02} .

Figure 4: $D=30$, ANOVA Test of Global Minimum for f_{02} .

Figure 5: $D = 30$, Evolution Curves of Fitness Value for f_{03} .

Figure 6: $D = 30$, ANOVA Test of Global Minimum for f_{03} .

Figure 7: $D = 30$, Evolution Curves of Fitness Value for f_{04} .

Figure 8: $D=30$, ANOVA Test of Global Minimum for f_{04} .

Figure 9: $D=30$, Evolution Curves of Fitness Value for f_{05} .

Figure 10: $D=30$, ANOVA Test of Global Minimum for f_{05} .

Figure 11: $D=30$, Evolution Curves of Fitness Value for f_{06} .

Figure 12: $D=30$, ANOVA Test of Global Minimum for f_{06} .

Figure 13: $D=30$, Evolution Curves of Fitness Value for f_{07} .

Figure 14: $D = 30$, ANOVA Test of Global Minimum for f_{07} .

Figure 15: $D = 30$, Evolution Curves of Fitness Value for f_{08} .

variance is relatively large. Although the results of f_{12} did not meet the standard, the variance is still ranked in the best of six, and the effect is more obvious. It is shown that the ECOA has better stability and robustness in the multimodal function optimization problem.

Figures 15–24 show the convergence and variance plots of the ECOA and other comparison algorithms in the f_{08} to f_{12} test functions. In f_{09} to f_{12} convergent graphs, the logarithm of the Y-axis is taken. In the f_{08} function, the logarithm is not taken because the theoretical optimal value is negative. It is easy to find that f_{08} to f_{12} converge faster than other algorithms, and in the variance graph, f_{08} to f_{12} are stable compared to other algorithms.

In Table 4, the ECOA finds the standard results from the comparison of f_{13} to f_{23} , the 11 low-dimensional test functions, and the overall ranking is also relatively high, where f_{13} , f_{14} , f_{16} , f_{18} , and f_{20} are in the variance rank. However, it still found the theoretical optimal value, and the variance is basically in the same series. Although FPA ranked first in f_{19} , it did not find the theoretical optimal value, but both the COA and the ECOA found the optimal value, and the variance and FPA also did not differ much. f_{15} , f_{17} , f_{21} , f_{22} , and f_{23} are ranked first, showing a stronger search capability, higher accuracy, and high robustness.

Figures 25–46 show the convergence and variance graphs of the ECOA and other comparison algorithms in f_{13} to f_{23} test functions, where f_{13} is logarithmic to the Y-axis, and it is easy to see that it converges faster than

Figure 16: $D=30$, ANOVA Test of Global Minimum for f_{08} .

Figure 17: $D=30$, Evolution Curves of Fitness Value for f_{09} .

Figure 18: $D=30$, ANOVA Test of Global Minimum for f_{09} .

Figure 19: $D = 30$, Evolution Curves of Fitness Value for f_{10} .

Figure 20: $D = 30$, ANOVA Test of Global Minimum for f_{10} .

Figure 21: $D = 30$, Evolution Curves of Fitness Value for f_{11} .

Figure 22: $D=30$, ANOVA Test of Global Minimum for f_{11} .

Figure 23: $D=30$, Evolution Curves of Fitness Value for f_{12} .

Figure 24: $D=30$, ANOVA Test of Global Minimum for f_{12} .

Figure 25: $D=4$, Evolution Curves of Fitness Value for f_{13} .

Figure 26: $D=4$, ANOVA Test of Global Minimum for f_{13} .

Figure 27: $D=2$, Evolution Curves of Fitness Value for f_{14} .

Figure 28: $D=2$, ANOVA Test of Global Minimum for f_{14} .

Figure 29: $D=2$, Evolution Curves of Fitness Value for f_{15} .

Figure 30: $D=2$, ANOVA Test of Global Minimum for f_{15} .

Figure 31: $D=2$, Evolution Curves of Fitness Value for f_{16} .

Figure 32: $D=2$, ANOVA Test of Global Minimum for f_{16} .

Figure 33: $D=3$, Evolution Curves of Fitness Value for f_{17} .

Figure 34: $D=3$, ANOVA Test of Global Minimum for f_{17} .

Figure 35: $D=6$, Evolution Curves of Fitness Value for f_{18} .

Figure 36: $D=6$, ANOVA Test of Global Minimum for f_{18} .

Figure 37: $D=4$, Evolution Curves of Fitness Value for f_{19} .

Figure 38: $D=4$, ANOVA Test of Global Minimum for f_{19} .

Figure 39: $D=4$, Evolution Curves of Fitness Value for f_{20} .

Figure 40: $D=4$, ANOVA Test of Global Minimum for f_{20} .

Figure 41: $D=4$, Evolution Curves of Fitness Value for f_{21} .

Figure 42: $D=4$, ANOVA Test of Global Minimum for f_{21} .

other algorithms. In f_{14} to f_{23} functions, the theoretical optimal value is negative, so there is no logarithm. The convergence speed of f_{14} to f_{20} , f_{23} is relatively fast compared to other contrast algorithms. In the ANOVA, f_{14} to f_{23} compared to other contrast algorithms are very stable.

Figure 43: $D=2$, Evolution Curves of Fitness Value for f_{22} .

Figure 44: $D=2$, ANOVA Test of Global Minimum for f_{22} .

Figure 45: $D=2$, Evolution Curves of Fitness Value for f_{23} .

Figure 46: $D = 2$, ANOVA Test of Global Minimum for f_{23} .

4.2 p -Values of the Wilcoxon Rank-Sum Test

In this section, the Wilcoxon rank-sum test [12, 26] is used to test the performance of the function, and the test results are distinguished by $p = 0.05$. $p > 0.05$ indicates that the result is accidental, and $p < 0.05$ indicates that the result is not accidental. Higher reliability underlined results are worse.

In Table 5, data with $p > 0.05$ are underlined. In comparison to ABC, the ECOA was significantly better. CS, FPA, and GWO, on the contrary, have data greater than 0.05. Although the COA has five more than 0.05, its overall view is good. Therefore, these results are not accidental, and we can see that the ECOA in the function test results are excellent.

4.3 ECOA for Engineering Optimization Problem

Design optimization, especially structural design optimization, has a wide range of applications in engineering and industry, because it has a lot of constraints. As a result, it is a test of its constraint to solve a problem. To verify the effectiveness of the algorithm for complex optimization problems, in this chapter, three engineering design examples are used: pressure vessel design problem [10], cantilever beam design problem [3], and welded beam design problem [5].

4.3.1 Pressure Vessel Design Problem

The pressure vessel design problem [10] is a classical hybrid constrained optimization problem. The working pressure is 2000 psi and the maximum capacity is 750 ft³. As shown in Figure 47, both ends of the cylindrical container are covered with a hemispherical head. Using a rolled steel plate, the shell is made into two halves, which are joined by two longitudinal welds to form a cylinder. The goal is to minimize the total cost, including the cost of materials, molding, and welding.

$$\text{Minimize } f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.8x_1^2x_3$$

$$\text{Subject to } g_1(x) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(x) = -x_3 + 0.00954x_3 \leq 0$$

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

$$g_4(x) = x_4 - 240 \leq 0$$

Table 5: p -Values of the Wilcoxon Rank-Sum Test Results.

Functions	ECOA vs. COA	ECOA vs. GWO	ECOA vs. FPA	ECOA vs. ABC	ECOA vs. CS
f_{01}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{02}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{03}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{04}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{05}	0.026077	0.001953	3.02E-11	3.02E-11	3.02E-11
f_{06}	3.2E-09	1.21E-12	0.182655	3.02E-11	3.02E-11
f_{07}	9.92E-11	0.000471	3.02E-11	3.02E-11	3.02E-11
f_{08}	2.00141E-10	2.78622E-11	2.75124E-11	5.77359E-07	2.78622E-11
f_{09}	N/A	0.333711	1.21E-12	1.21E-12	1.21E-12
f_{10}	N/A	8.6442E-14	1.21178E-12	1.21178E-12	1.21178E-12
f_{11}	N/A	6.60964E-05	1.21178E-12	1.21178E-12	1.21178E-12
f_{12}	2.60151E-08	3.01986E-11	3.01986E-11	3.01986E-11	3.01986E-11
f_{13}	0.534244	1.96E-10	5.09E-10	2.73E-11	5.09E-10
f_{14}	N/A	1.21E-12	0.041774	2.71E-14	N/A
f_{15}	N/A	N/A	N/A	1.21178E-12	5.82631E-09
f_{16}	0.100331	1.09E-11	0.000192	1.08E-11	0.502352
f_{17}	N/A	1.21178E-12	N/A	9.65378E-13	N/A
f_{18}	0.367771	9.53E-10	9.53E-10	3.42E-08	8.64E-08
f_{19}	0.160742	1.21E-12	1.69E-14	1.14E-12	5E-11
f_{20}	0.790214	1.01E-11	4.28E-13	2.39E-11	1.11E-10
f_{21}	N/A	1.21178E-12	6.13374E-14	1.19214E-12	1.14996E-12
f_{22}	N/A	1.21E-12	1.69E-14	0.000661	N/A
f_{23}	N/A	0.011035	N/A	1.21E-12	1.21E-12

Figure 47: Pressure Vessel Design Problem.

In this paper, the ECOA is used to solve this problem, in which the comparison algorithms run 20 times independently, with GA [8], HS [14], ABC [18], CS [10], GSA [22], CoBiDE [25], DSA [4], and AMO [15] are presented in Table 6.

The ECOA is less costly than other comparison algorithms. Thus, the ECOA used to solve the pressure vessel design problem is feasible.

Table 6: Comparison Results for the Pressure Vessel Design Problem.

Algorithm	Optimal values for variables				Optimal cost
	x1	x2	x3	x4	
GSA [22]	1.125000	0.625000	55.9886598	84.4542025	8538.8359
GA [8]	0.937500	0.500000	48.329000	112.679000	6410.3811
HS [14]	1.125000	0.625000	58.278900	43.75490000	7198.433
ABC [18]	0.8337011	0.41792373	43.0948918	164.781043	6021.770461
CS [10]	0.812500	0.437500	42.0984456	176.6365958	6059.7143348
CoBiDE [25]	0.7781686	0.38464916	40.3196187	199.999998	5885.332773
DSA [4]	0.7826957	0.38531758	40.3859923	199.881092	5928.486479
AMO [8]	0.7850776	0.38917280	40.6371423	195.973315	5913.45051
ECOA	0.7781686	0.3846492	40.31962	199.999998	5885.332773

Figure 48: Cantilever Beam Design Problem.

4.3.2 Cantilever Beam Design Problem

The cantilever beam design problem [11] structure is shown in Figure 48, which minimizes the weight of the cantilever beam with a hollow square block. There are five squares, where the first block is fixed and the fifth block is borne by the vertical load. The five parameters define the shape of the cross-section of the cube. The form of the problem is as follows:

$$\text{Minimize } f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5);$$

$$\text{Subject to } g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1;$$

In this paper, the ECOA is used to solve this problem. The algorithms include the method of moving asymptotes (MMA) [5], generalized convex approximation (GCA_I) [5], GCA_II [5], CS [10], symbiotic organisms search (SOS) [2], and MVO [20] run independently 20 times, and the results are presented in Table 7.

It can be found by comparison that the minimum cost of the ECOA in solving the cantilever problem is in the first place, which can verify the superiority of the algorithm.

4.3.3 Welded Beam Design Problem

The welded beam design problem is a problem that is academically widely used in practical engineering design (Figure 49). The purpose of the design is to minimize the manufacturing costs. The design model

Table 7: Comparison Results for the Cantilever Beam Design Problem.

Algorithm	Optimal values for variables					Optimal cost
	x1	x2	x3	x4	x5	
MMA [5]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_I [5]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II [5]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [10]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [2]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
MVO [8]	6.02394	5.30601	4.49501	3.49602	2.15273	1.3399595
ECOA	6.015957	5.309176	4.4943367	3.5015356	2.1526533	1.3399564

is affected by the shear stress (τ), the beam bending stress (θ), the rod buckling load (P_c), the beam end deflection (δ), and the normal stress (σ) related to the seven conditional constraints. Specific optimization problems can be described as a formula. The four decision variables x_1 to x_4 defined the domain as the formula:

$$\text{Minimize } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2);$$

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

$$g_2(x) = \sigma(x) - \sigma_{\max} \leq 0,$$

$$g_3(x) = x_3 - x_4 \leq 0,$$

$$g_4(x) = 0.125 - x_1 \leq 0,$$

$$g_5(x) = \delta(x) - 0.25 \leq 0,$$

$$g_6(x) = P - P_c(x) \leq 0,$$

$$g_7(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0;$$

$$\text{Variable range } 0.1 \leq x_1 \leq 2; 0.1 \leq x_2 \leq 10; 0.1 \leq x_3 \leq 10; 0.1 \leq x_4 \leq 2;$$

$$\text{where } \tau(x) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2};$$

$$\tau_1 = \frac{P}{\sqrt{2}x_1x_2};$$

$$\tau_2 = \frac{MR}{J};$$

$$M = P\left(L + \frac{x_2}{2}\right);$$

$$J(x) = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_2}{2}\right)^2\right]\right\};$$

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

$$\sigma(x) = \frac{6PL}{x_4x_3^2};$$

$$\delta(x) = \frac{6PL^3}{Ex_3^3x_4};$$

Figure 49: Welded Beam Design Problem.

$$P_c = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right).$$

In this paper, the ECOA is used to solve this problem, which includes GWO [19], GSA [19], CPSO [19], GA (Coello) [10], GA (Deb) [9], GA (Deb) [7], HS [5], Random [21], Simplex [21], David [21], APPROX [21], and BA [11] run 20 times independently and presented in Table 8.

By comparison, we can find that the ECOA has the lowest cost and the most obvious effect in the welded beam design problem, which shows that the ECOA has obvious advantages in the welded beam design problem.

4.4 Result Analysis

In Section 4.1, 23 standard benchmark functions were selected to evaluate the performance of the ECOA. f_{01} to f_{07} are unimodal functions, f_{08} to f_{12} are multimodal functions, and f_{13} to f_{23} are low-dimensional functions. The experimental results are shown in Tables 2–4. Figures 1–46 show the convergence and variance graphs of the corresponding functions. The results in the table show that the ECOA can find a more accurate solution. Convergence maps and variance maps reflect that the ECOA has faster convergence and higher stability. In

Table 8: Comparison Results for the Welded Beam Design Problem.

Algorithm	Optimal values for variables				Optimal cost
	h	l	t	b	
GWO [19]	0.205676	3.478377	9.03681	0.205778	1.72624
GSA [19]	0.182129	3.856979	10.0000	0.202376	1.87995
CPSO [19]	0.202369	3.544214	9.048210	0.205723	1.72802
GA (Coello) [10]	N/A	N/A	N/A	N/A	1.8245
GA (Deb) [6]	N/A	N/A	N/A	N/A	2.3800
GA (Deb) [24]	0.2489	6.1730	8.1789	0.2533	2.4331
HS [5]	0.2442	6.2231	8.2915	0.2443	2.3807
Random [21]	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex [21]	0.2792	5.6256	7.7512	0.2796	2.5307
David [21]	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX [21]	0.2444	6.2189	8.2915	0.2444	2.3815
BA [11]	0.2015	3.562	9.0414	0.2057	1.7312
ECOA	0.20573	3.25312	9.036624	0.20573	1.695247

Section 4.3, three structural design issues (pressure vessel design problem, welded beam design problem, and cantilever design problem) were selected to test the proposed ECOA. The results show that the ECOA performs well in solving constrained optimization problems.

5 Conclusions

In this paper, the ECOA is proposed based on the COA, which is used to solve the problem of functional optimization and structural engineering design. By introducing the elite opposite learning strategy, it can help to improve its exploration ability. From the results of the 23 benchmark functions and three engineering design problems, the performance of the ECOA is superior to that of other group-based intelligent algorithms mentioned in this paper. Compared to other algorithms, the ECOA is faster and more accurate at the convergence rate, whereas the variance is smaller than the others to show its stability. In addition, it can be seen that the ECOA is more robust, so its development prospects are still relatively broad.

In future research, we hope to combine the cultural algorithm by combining the two-layer mechanism of cultural algorithm to produce a three-layer structure of cultural cognitive algorithm to improve the performance of cultural algorithms. At the same time, the COA will be applied to the large-scale 0-1 backpack, which is the traditional NP-hard problem to reflect the searchability and robustness of the algorithm. In addition, through the introduction of learning strategies and other excellent strategies to improve the algorithm itself, to solve more examples of life, such as the fields of wireless sensor network coverage optimization problem and the image edge detection. Through the in-depth study of the algorithm to expand the scope of application of the cognitive behavior algorithm.

Acknowledgments: This work is supported by the National Natural Science Foundation of China grant nos. 61463007 and 61563008 and the Project of Guangxi University for Nationalities Science Foundation grant 2012MDZD037.

Bibliography

- [1] B. Basturk and D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, in: *IEEE Swarm Intelligence Symposium*, Indiana, 2006.
- [2] M.-Y. Cheng and D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, *Comput. Struct.* **139** (2014), 98–112.
- [3] H. Chickermane and H. C. Gea, Structural optimization using a new local approximation method, *Int. J. Numer. Methods Eng.* **39** (1996), 829–846.
- [4] P. Civicioglu, Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm, *Comput. Geosci. U. K.* **46** (2012), 229–247.
- [5] C. A. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, *Comput. Ind.* **41** (2000), 113–127.
- [6] M. Crepinšek, S.-H. Liu and M. Mernik, Replication and comparison of computational experiments in applied evolutionary computing: common pitfalls and guidelines to avoid them, *Appl. Soft Comput.* **19** (2014), 161–170.
- [7] K. Deb, Optimal design of a welded beam via genetic algorithms, *AIAA J.* **29** (1991), 2013–2015.
- [8] K. Deb and A. S. Gene, A robust optimal design technique for mechanical component design, in: D. Dasgupta and Z. Michalewicz, (eds.), *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, pp. 497–514, 1997.
- [9] K. Deb, An efficient constraint handling method for genetic algorithms, *Comput. Methods Appl. Mech. Eng.* **186** (2000), 311–338.
- [10] A. H. Gandomi, X.-S. Yang and A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.* **29** (2013), 17–35.
- [11] A. Gandomi, X. S. Yang, A. Alavi and S. Talatahari, Bat algorithm for constrained optimization tasks, *Neural Comput. Appl.* **22** (2013), 1239–1255.

- [12] S. Garcia, D. Molina, M. Lozano and F. Herrera, A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, *J. Heuristics* **15** (2008), 617–644.
- [13] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
- [14] K. S. Lee and Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods Appl. Mech. Eng.* **194** (2005), 3902–3933.
- [15] X. Li, J. Zhang and M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, *Neural Comput. Appl.* **24** (2014), 1867–1877.
- [16] M. Li, H. Zhao, X. Weng and T. Han, Cognitive behavior optimization algorithm for solving optimization problems, *Appl. Soft Comput.* **39** (2016), 199–222.
- [17] J. J. Liang, B. Y. Qu and P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 Special Session and Competition on Single Objective Realparameter Numerical Optimization, Technical Report, 2013, pp. 1–32.
- [18] M. Mernik, S. H. Liu, M. D. Karaboga and M. Crepinšek, On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering anew implementation, *Inf. Sci.* **291** (2015) 115–127.
- [19] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* **69** (2014), 46–61.
- [20] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* **27** (2016), 495–513.
- [21] K. Ragsdell and D. Phillips, Optimal design of a class of welded structures using geometric programming, *ASME J. Eng. Ind.* **98** (1976), 1021–1025.
- [22] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, GSA: a gravitational search algorithm, *Inf. Sci.* **179** (2009), 2232–2248.
- [23] R. Storn and K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* **11** (1997), 341–359.
- [24] H. R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: *Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation*, IEEE, USA, pp. 695–701, 2005.
- [25] Y. Wang, H. X. Li, T. W. Huang and L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, *Appl. Soft Comput.* **18** (2014), 232–247.
- [26] F. Wilcoxon, Individual comparisons by ranking methods, *Biom Bull. Biomet.* **1** (1945), 80–83.
- [27] X. S. Yang and S. Deb, Cuckoo search via Levy flights, in: *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, IEEE Publication, USA, pp. 210–214, 2009.
- [28] X. S. Yang, Flower pollination algorithm for global optimization, in: *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, vol. 7445, 2012, pp. 240–249.
- [29] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier B.V., Amsterdam, Netherlands, 2014.
- [30] Y. Zhao and X.-S. Yang, *New Meta-heuristic Optimization Methods*, Science Press, Beijing, 2013.