Knowl Inf Syst (2019) 58:209–248
https://doi.org/10.1007/s10115-018-1158-1

REGULAR PAPER

Complex-valued encoding symbiotic organisms search algorithm for global optimization

Fahui Miao[1] · **Yongquan Zhou**[1,2] · **Qifang Luo**[1,2]

Received: 19 January 2017 / Revised: 19 October 2017 / Accepted: 5 January 2018 /
Published online: 16 January 2018

Abstract Symbiotic organisms search algorithm is a new meta-heuristic algorithm based on the symbiotic relationship between the biological which was proposed in recent years. In this paper, a novel complex-valued encoding symbiotic organisms search (CSOS) algorithm is proposed. The algorithm introduces the idea of complex coding diploid. Each individual is composed of real and imaginary parts and extends the search space from one dimension to two dimensions. This increases the diversity of the population, further enhances the ability of the algorithm to find the global optimal value, and improves the precision of the algorithm. CSOS has been tested with 23 standard benchmark functions and 2 engineering design problems. The results show that CSOS has better ability of finding global optimal value and higher precision.

Keywords Symbiotic organisms search · Complex-valued encoding · Benchmark test functions · Engineering problems

1 Introduction

Swarm intelligence optimization algorithm comes from simulating the behavior of various groups in nature, human society and animals. The purpose of finding the global optimal value is to use the individual information interaction and cooperation in the group. Compared with other types of optimization algorithms, swarm intelligence optimization algorithm is simple, easy to implement, higher efficiency and accuracy. At present, the most popular swarm intelligence optimization algorithms are ant colony optimization (ACO) [1], differential evolution (DE) [2], particle swarm optimization (PSO) [3]. In recent years, some new swarm intelligent

✉ Yongquan Zhou
yongquanzhou@126.com

1 College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

2 Key Laboratories of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, Guangxi, China

algorithms have been proposed, such as flower pollination algorithm (FPA) [4], cuckoo search (CS) [5], firefly algorithm (FA) [6], charged system search (CSS) [7], bat algorithm (BA) [8], grey wolf optimization (GWO) [9]. At present, swarm intelligent optimization algorithm, as a meta-heuristic algorithm based on swarm intelligence, has been widely used in many fields such as engineering, network communication, finance, automatic control and so on.

Symbiotic Organisms Search (SOS) was a new meta-heuristic algorithm proposed by Cheng and Prayogo [10]. Compared to most meta-heuristic algorithms, the SOS algorithm has an obvious advantage that the algorithm does not require special algorithm parameter settings. SOS algorithm structure is simple, easy to understand, so by more and more scholars of the study. At present, the symbiotic organism search algorithm has been applied to such as task scheduling in cloud computing environment [11], large-scale economic dispatch problem with valve-point effects [12], optimal power flow of power system with FACTS devices [13], DG placement in radial distribution network [14] and many other aspects.

In expressing neural network weights [15] and representing individual genes for evolutionary algorithms [16], the complex encoding [17] method has been applied. So this paper presents a complex-valued encoding symbiotic organisms search (CSOS). The original SOS algorithm is implemented in real-coded way to encode the algorithm. In this way, the application scope of the algorithm is limited to the real range, which limits the diversity of the population and is not conducive to the optimization of the algorithm. Compared with the real number coding, complex code has many advantages [15–17]. The contribution of this paper is to introduce the idea of complex coding into the SOS algorithm and propose a symbiotic organisms search algorithm based on complex coding. In the CSOS algorithm, the structure of the real and imaginary parts of the complex code is introduced into the SOS algorithm, and the two-dimensional coding space of the complex code is used to map the real-coded one-dimensional coding space. We use real and imaginary parts to collectively represent a biological individual in the population, and the real and imaginary parts are updated separately to find the optimal value of the algorithm. This haploid structure expands the information contained in the individual genes of the organism in the symbiotic organisms search algorithm, increases the biodiversity of the individual in the population, improves the possibility of obtaining the optimal solution, and enhances the optimization of the algorithm ability.

The remainder of this paper is structured as follows: Sect. 2 briefly introduces the basic symbiotic organism search (SOS) algorithm; Sect. 3 presents a complex-valued encoding symbiotic organisms search (CSOS) algorithm; simulation experiments and results analysis are presented in Sect. 4; Sect. 5 presents the conclusions of this paper.

2 Symbiotic organisms search (SOS)

Symbiotic Organism Search (SOS) was proposed by Cheng and Prayogo [10]. The SOS algorithm is inspired by the interaction between various organisms in an ecosystem. In nature, biological individuals usually use the symbiotic relationship with other organisms to improve their survival ability. In an ecosystem, mutualism, commensalism, and parasitism are the most fundamental relationships found in the living organisms. These three symbiotic relationships are shown in Fig. 1 [18]. The details about these processes are narrated below [10,14].

Fig. 1 Process of 'Symbiosis' among natural organisms in an ecosystem

2.1 Mutualism phase

This phase the interaction between two different organisms provide benefits to both of them. As shown in Fig. 1, the relationship between flower and pollinator is a classic example to explain the philosophy of mutualism.

In SOS, X_i is an organism (matched to the ith member of the ecosystem) that interacts with another randomly selected organism X_j from the ecosystem. Both the organisms are engaged in mutualism relationship with the goal of increasing their mutual survival advantage in the ecosystem. The new solution, after the mutualism phase for X_{inew} and X_{jnew}, which is modeled in Eqs. (1) and (2),

$$X_{inew} = X_i + rand(0, 1) * (X_{\text{best}} - Mutual\_Vector * \text{BF}_1) \quad (1)$$

$$X_{jnew} = X_j + rand(0, 1) * (X_{\text{best}} - Mutual\_Vector * \text{BF}_2) \quad (2)$$

$$Mutual\_Vector = \frac{X_i + X_j}{2} \quad (3)$$

In ecosystems, the benefits of mutualism relationship may be unequal from each other. The benefit factors (BF_1 and BF_2) are randomly chosen 1 or 2. *Mutual_Vector* represents the relationship between the two biological X_i and X_j.

2.2 Commensalism phase

The Commensalism phase is the relationship between the two random organisms which one to gain benefit, while the other one has no effect. The most common examples of commensalism relationships in nature are sharks and remora fish. The remora fish is usually absorbed on the shark and depends on the remaining food residue to survive. In this relationship, the remora fish unilaterally gets the benefit, while the shark does not affect.

In SOS, X_i from the ecosystem were randomly selected with a X_j composed of a mutualism relationship. Only X_i single side benefit from X_j. According to the above rules, X_i update formula as (4).

$$X_{inew} = X_i + rand(-1, 1) * (X_{\text{best}} - X_j) \quad (4)$$

2.3 Parasitism phase

In the parasitism phase, one organism randomly chooses another organism to establish a parasitism relationship. In this parasitism relationship, one organism benefits from another, and the other are the victims. The most common examples of parasitism relationships in nature are anopheles mosquito and human host.

In SOS, X_i by creating an artificial parasite called as Parasite_Vector to play the role of anopheles mosquito. Parasite_Vector was created by duplicating organism X_i, then modifying the randomly selected dimensions using a random number. The organism X_j is randomly selected from the ecosystem and is used as a host. By comparing the fitness value of X_j and Parasite_Vector in the ecosystem, the better one will survive, while the other with low value that will be eliminated.

Algorithm 1. Symbiotic Organism Search Algorithm

Initialize a population of n organisms(Number of organisms in the eco system) with random solutions
Identify the best organism(X_{best}) in the initial population
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
While($t < MaxGeneration$)
  for *i=1:n* (Number of organisms in the eco system)
    Mutualism Phase
      Choose organism j randomly other than organism i
      Determine Beneficial Factor and mutual Vector via Eqs.(3)
      Modify organism X_i and X_j based on their mutual relationship via Eqs.(1) and (2)
      Calculate new solution after Mutualism Phase
      Evaluate the fitness of the new solution
      Accept the new solution if the fitness is better
    End of Mutualism Phase
    Commensalism Phase
      Choose organism j randomly other than organism i
      Modify organism X_i with the assist of organism X_j via Eq. (4)
      Calculate new solution after Commensalism Phase
      Evaluate the fitness of the new solution
      Accept the new solution if the fitness is better
    End of Commensalism Phase
    Parasitism Phase
      Choose organism j randomly other than organism i
      Create a Parasite ($Parasite\_Vector$) from Organism X_i
      Calculate Fitness Value of the new organism
      Kill organism j and replace it with the parasite if the fitness is lower than the parasite
    End of Parasitism Phase
    Update the best organism
  end for
  $t = t + 1$
end while
Output the global optimization solution.

Table 1 Symbiotic organisms chromosome model

$Gene_1$	$Gene_2$	$Gene_i$	$Gene_M$
(R_{P1}, I_{p1})	(R_{P2}, I_{p2})	…	(R_{PM}, I_{pM})

3 Complex-valued encoding symbiotic organisms search (CSOS)

In nature, the chromosome of complex biological tissue is generally provided by the parent body, each of which is provided with a pair of chromosomes. Because of the two-dimensional nature of complex coding, it is natural to use this to represent a pair of chromosomes in the allele. The real and imaginary parts of complex numbers are called real genes and virtual genes. For a problem with M independent variables, the complex representation is shown in Eq. (5).

$$x_p = R_p + il_p \qquad p = 1, 2, 3, \ldots, M. \tag{5}$$

The gene of the organism can be expressed as a diploid structure and recorded as (R_p, I_p). Where R_p and I_p represent the real and imaginary parts of the complex number, respectively. Thus, the chromosomal model of the organism can be represented as shown in the following Table 1.

3.1 Initializing the complex-valued encoding population

According to the definition interval $[A_k, B_k]$, $k = 1, 2, \ldots M$, of the problem, M modules and M amplitudes [16] are randomly generated:

$$\rho_k = \left[0, \frac{B_k - A_k}{2}\right], \quad k = 1, 2, \ldots M \tag{6}$$

$$\theta_k = [-2\pi, 2\pi], \quad k = 1, 2, \ldots M \tag{7}$$

According to formula (8) we get M complex numbers:

$$X_{Rk} + iX_{Ik} = \rho_k(\cos\theta_k + i\sin\theta_k), \quad k = 1, 2, \ldots M \tag{8}$$

Through the above process, we can get M real part and M imaginary part at the same time and then update them, respectively, in the following way.

3.2 The updating method of CSOS

3.2.1 Mutualism phase

(1) Update the Real Parts:

$$X_R(i+1) = X_R(i) + rand(0, 1) * (X_{Rbest} - Mutual\_Vector_R * \mathrm{BF}_1) \tag{9}$$

$$X_R(j+1) = X_R(j) + rand(0, 1) * (X_{Rbest} - Mutual\_Vector_R * \mathrm{BF}_2) \tag{10}$$

$$Mutual\_Vector_R = \frac{X_R(i) + X_R(j)}{2} \tag{11}$$

(2) Update the Imaginary Parts

$$X_I(i+1) = X_I(i) + rand(0,1) * (X_{Ibest} - Mutual\_Vector_I * \ BF_1) \tag{12}$$

$$X_I(j+1) = X_I(j) + rand(0,1) * (X_{Ibest} - Mutual\_Vector_I^* \ \mathrm{BF}_2) \tag{13}$$

$$Mutual\_Vector_I = \frac{X_I(i) + X_I(j)}{2} \tag{14}$$

where X_{Rbest} and X_{Ibest} represent the optimal solution of real and imaginary parts of all living organisms in the whole symbiotic population. $Mutual\_Vector_R$ and $Mutual\_Vector_I$ represent the real and imaginary parts of the two biological relationships, respectively.

3.2.2 Commensalism phase

(1) Update the Real Parts:

$$X_R(i+1) = X_R(i) + rand(-1,1) * (X_{Rbest} - X_R(j)) \tag{15}$$

(2) Update the Imaginary Parts

$$X_I(i+1) = X_I(i) + rand(-1,1) * (X_{Ibest} - X_I(j)) \tag{16}$$

3.2.3 Parasitism phase

(1) Update the Real Parts:

In SOS, $X_R(i)$ by creating an artificial parasite called as *Parasite_Vector*$_R$ to play the role of anopheles mosquito. *Parasite_Vector*$_R$ was created by duplicating organism $X_R(i)$, then modifying the randomly selected dimensions using a random number.

(2) Update the Imaginary Parts

Similarly, $X_I(i)$ by creating an artificial parasite called as *Parasite_Vector*$_I$ to play the role of anopheles mosquito. *Parasite_Vector*$_I$ was created by duplicating organism, then modifying the randomly selected dimensions using a random number.

3.3 The calculation method of fitness value

Because the complex number is composed of two parts: the real part and the imaginary part, we need to transform the coding space in the computation of fitness [16]. Therefore, before calculating the fitness value, we need to convert the complex number to real number and then calculate the fitness function value. The concrete practices are as follows:

$$\rho_n = \sqrt{X_{R_n}^2 + X_{I_n}^2}, \quad n = 1, 2, \ldots, M. \tag{17}$$

$$RV_n = \rho_n \mathrm{sgn}\left(\sin\left(\frac{X_{I_n}}{\rho_n}\right)\right) + \frac{B_k + A_k}{2}, \quad n = 1, 2, \ldots, M. \tag{18}$$

where RV_n is the real variable argument after conversion. According to the real variable, the corresponding fitness function value is calculated and evaluated. If it is better than the current optimal value, it is replaced. Otherwise, the next iteration is carried out.

3.4 CSOS algorithm pseudo code

The CSOS is to incorporate the two-dimensional idea of complex number into it. In CSOS, the real part and the imaginary part are updated, respectively, which enriches the diversity of the population and enhances the global searching ability of the individual in the algorithm, and improves the performance of the algorithm.

Algorithm 2. Complex-Valued Encoding Organism Search Algorithm

Initialize a population of n organisms: $\rho_k = [0, \frac{B_k - A_k}{2}]$ and $\theta_k = [-2\pi, 2\pi]$

Get the real and imaginary part of the complex [Eq. (8)]

Convert to real variables [Eq. (17) and Eq. (18)]

Identify the best organism in the initial population

Define a stopping criterion (either a fixed number of generations/iterations or accuracy)

While ($t < MaxGeneration$ **)**

  for *i=1:n* (Number of organisms in the eco system)

    Mutualism Phase

      Choose organism j randomly other than organism i

       Update the real part [Eq. (9), Eq. (10) and Eq. (11)]

       Update the imaginary part [Eq. (12), Eq. (13) and Eq. (14)]

       Convert to real variables [Eq. (17) and Eq. (18)]

       Calculate new solution after Mutualism Phase

       Evaluate the fitness of the new solution

       Accept the new solution if the fitness is better

    End of Mutualism Phase

    Commensalism Phase

      Choose organism j randomly other than organism i

      Update the real part [Eq. (15)]

      Update the imaginary part [Eq. (16)]

      Convert to real variables [Eq. (17) and Eq. (18)]

      Calculate new solution after Commensalism Phase

      Evaluate the fitness of the new solution

      Accept the new solution if the fitness is better

    End of Commensalism Phase

    Parasitism Phase

      Choose organism j randomly other than organism i

      Create a Parasite ($Parasite\_Vector_R$) from Organism $X_R(i)$

      Create a Parasite ($Parasite\_Vector_I$) from Organism $X_I(i)$

      Convert to real variables [Eq. (17) and Eq. (18)]

      Calculate Fitness Value of the new organism

      Kill organism j and replace it with the parasite if the fitness is lower than the parasite

    End of Parasitism Phase

    Update the best organism

  end for

  $t = t + 1$

end while

Output the best solution.

4 Simulation experiments and result analysis

To verify the effectiveness and superiority of Complex-Valued Encoding Organism Search Algorithm (CSOS), the test of 23 standard test functions [19,20] were tested. These 23 standard test functions are widely used in the literature. Section 4.1 gives the environment configuration of the simulation experiment. Section 4.2 Comparison results of performance of each algorithm are given. Section 4.3 The Wilcoxon rank-sum test results for CSOS and several other algorithms are given.Section 4.4 CSOS is applied to the cantilever beam and welding beam two engineering optimization problems.

4.1 Experimental setup

The development environment for this test is MATLAB R2012a. The test runs on AMD Athlont (tm) II*4640 processor and 4 GB memory.

4.2 Comparison of each algorithm performance

The CSOS algorithm proposed in this paper is compared with the mainstream group intelligent optimization algorithm ABC [1], CS [5], FPA [4], GWO [9], CGWO [22], SOS [10] from four aspects: the best value, the worst value, the average value and the standard. The control parameters involved in the above algorithm are shown below.

ABC setting: limit $= 5D$ has been used as recommended in [21], the population size is 20. The maximum iteration number is 100.

CS setting: $\beta = 1.5$, $\rho_0 = 1.5$ have been used as recommended in [5], the population size is 20. The maximum iteration number is 100.

FPA setting: switch probability $\rho = 0.8$ in accordance with the suggestions given in [10], the population size is 20. The maximum iteration number is 100.

GWO setting: $\vec{\alpha}$ Linearly decreased from 2 to 0 have been used as recommended in [9], the population size is 20. The maximum iteration number is 100.

CGWO setting: $\vec{\alpha}$ Linearly decreased from 2 to 0 have been used as recommended in [22], the population size is 20. The maximum iteration number is 100.

SOS setting: the population size is 20. The maximum iteration number is 100.

In this paper, the fifteen independent tests of the three standard benchmark functions (unimodal benchmark functions, multimodal benchmark functions, fixed-dimension multimodal benchmark functions) in Tables 2, 3 and 4 were carried out. The results of unimodal, multimodal and fixed-dimension multimodal are shown in Tables 5, 6 and 7, respectively. In the table, the best fitness value, the worst fitness value, the average fitness value and the standard deviation in the experiment are, respectively, expressed by Best, Worst, Mean and Std. All algorithms are ranked according to the value of std.

According to the test results obtained in Table 5, only the CSOS algorithm finds the theoretical optimal value zero of the unimodal benchmark functions f_1, f_2, f_3, f_4. This shows that compared with other algorithms, CSOS has a stronger ability to find the minimum. According to the mean value and the variance, we can see that it has high robustness in unimodal benchmark functions f_1, f_2, f_3, f_4. f_5 and f_6 are only slightly worse than the SOS algorithm in finding global minimum values, but CSOS values are smaller and more stable in terms of variance. CSOS in f_7 to find the minimum is less than other algorithms. In addition, the standard deviation of CSOS is the least, which indicates that it has more stability than other algorithms. Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15 shows CSOS and other algorithm convergence and the anova tests of the global minimum plots. It can be seen from

Table 2 Unimodal benchmark functions

No.	Name	Benchmark function	Dim	Scope	$f_{\min}$
f_1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	30	$x_i \in [-100, 100]$	0
f_2	Schwefel2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$x_i \in [-10, 10]$	0
f_3	Schwefel1.2	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$x_i \in [-100, 100]$	0
f_4	Schwefel2.21	$f(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le D \}$	30	$x_i \in [-100, 100]$	0
f_5	Resonbrock	$f(x) = \sum_{i=1}^{D-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30	$x_i \in [-30, 30]$	0
f_6	Step	$f(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	$x_i \in [-100, 100]$	0
f_7	Quartic	$f(x) = \sum_{i=1}^{n} x_i^4 + random(0, 1)$	30	$x_i = [-1.28, 1.28]$	0

Table 3 Multimodal benchmark functions

No.	Name	Benchmark function	Dim	Scope	$f_{\min}$
f_8	Rastrigin	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$x_i \in [-5.12, 5.12]$	0
f_9	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)\right) + 20 + e$	30	$x_i \in [-32, 32]$	0
f_{10}	Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n}\left(x_i^2\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$x_i \in [-600, 600]$	0
f_{11}	Penalty#2	$f(x) = 0.1\left\{\sin^2\left(3\pi x_1 + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right)\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$x_i \in [-50, 50]$	0

Table 4 Fixed-dimension multimodal benchmark functions

No.	Name	Benchmark function	Dim	Scope	$f_{\min}$
f_{12}	Shekel's Foxholes	$f(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	$x_i\in[-65,65]$	1
f_{13}	Kowalik	$f(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	$x_i\in[-5,5]$	0.0003075
f_{14}	Six hump camel back	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$x_i\in[-5,5]$	− 1.0316285
f_{15}	Drop wave	$f(x)=-\frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{0.5\left(x_1^2+x_2^2\right)+2}$	2	$x_i\in[-5.12,5.12]$	− 1
f_{16}	GoldStein Price	$f(x)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	$x_i\in[-5,5]$	3
f_{17}	Hartman	$f(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	$x_i\in[1,3]$	− 3.86
f_{18}	Hartman	$f(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	$x_i\in[0,1]$	− 3.32
f_{19}	Shekel 1	$f(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	$x_i\in[0,10]$	− 10.1532
f_{20}	Shekel 2	$f(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	$x_i\in[0,10]$	− 10.4029

Table 4 continued

No.	Name	Benchmark function	Dim	Scope	$f_{\min}$
f_{21}	Shekel 3	$f(x) = -\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^T + c_i\right]^{-1}$	4	$x_i \in [0, 10]$	-10.5364
f_{22}	Easom	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	2	$x_i \in [-100, 100]$	-1
f_{23}	Schaffer	$f(x) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	$x_i \in [-100, 100]$	-1

Table 5 Simulation results for test functions $f_i, i = 1, 2, 3, 4, 5, 6, 7$

Benchmark functions	Result	Method							Rank
		ABC	CS	FPA	GWO	CGWO	SOS	CSOS	
$f_1(D = 30)$	Best	13,078.16	3029.721	7.41505	0.015217	6.31E−10	1.01E−25	0	1
	Worst	34,156.9	6673.182	15.63149	0.261259	3.27E−05	2.18E−23	0	
	Mean	22,630.22	4733.173	10.39877	0.100055	9.78E−06	2.69E−24	0	
	Std	5588.282	1144.164	2.260591	0.07121	1.28E−05	5.42E−24	0	
$f_2(D = 30)$	Best	46.04829	47.6031	11.03045	0.030303	7.56E−05	9.86E−14	0	1
	Worst	44,207.93	64,368.57	16.57293	0.135566	0.001983	1.06E−12	0	
	Mean	3001.229	4360.252	13.75543	0.075162	0.000713	4.27E−13	0	
	Std	11,399.49	16,600.81	1.754746	0.030504	0.00048	2.78E−13	0	
$f_3(D = 30)$	Best	44,955.43	8651.311	7.167018	291.7691	0.055652	1.37E−07	0	1
	Worst	96,362.73	28,086.28	25.78581	3053.987	27,288.63	4.76E−05	0	
	Mean	68,236.05	20,031.69	17.1145	787.5964	3177.004	4.33E−06	0	
	Std	13,551.78	4463.591	4.436065	675.6151	7111.002	1.21E−05	0	
$f_4(D = 30)$	Best	84.41232	28.49209	1.035495	1.759882	0.017146	5.95E−11	0	1
	Worst	96.34205	43.50404	1.726446	5.137366	0.896027	8.71E−10	0	
	Mean	89.78934	35.819	1.371201	2.819192	0.251988	2.85E−10	0	
	Std	3.695347	4.015917	0.204814	0.927166	0.228299	1.89E−10	0	
$f_5(D = 30)$	Best	37,443,448	90,3924.9	1129.815	30.00032	27.557	25.2798	28.02087	1
	Worst	1.22E+08	5,324,893	3244.24	88.40484	29.93857	27.78508	28.9901	
	Mean	77,364,397	2,242,871	1793.053	38.70693	28.72041	26.53727	28.82546	
	Std	21,304,161	1,240,804	559.3148	14.04453	0.528622	0.618768	0.323658	

Table 5 continued

Benchmark functions	Result	Method							Rank
		ABC	CS	FPA	GWO	CGWO	SOS	CSOS	
$f_6(D = 30)$	Best	17,365.68	2561.882	7	2.871506	0.00043	0.001149	4.250911	3
	Worst	38,531.03	5694.885	23	5.397938	1.607856	0.09064	7	
	Mean	24,180.84	4221.635	12.26667	4.155023	0.324767	0.012878	5.399195	
	Std	5489.721	922.6223	3.918211	0.830692	0.414991	0.021962	0.801052	
$f_7\ (D = 30)$	Best	10.04376	0.435972	50.60509	0.007633	0.001191	0.00138	3.24E−05	1
	Worst	55.26342	2.186526	275.4641	0.045352	0.026963	0.006376	0.000995	
	Mean	27.45216	1.446428	138.3756	0.023563	0.011492	0.00311	0.000411	
	Std	11.50779	0.557719	74.10931	0.010323	0.008559	0.001497	0.00028	

Table 6 Simulation results for test functions $f_i, i = 8, 9, 10, 11$

Benchmark functions	Result	Method							Rank
		ABC	CS	FPA	GWO	CGWO	SOS	CSOS	
$f_8(D = 30)$	Best	186.2117	195.8951	39.59563	13.19795	1.36E−06	0	0	1
	Worst	289.6676	260.5012	117.0963	57.24281	155.4529	0	0	
	Mean	232.9944	229.023	79.8071	30.54779	34.35652	0	0	
	Std	29.89989	17.39867	31.53451	11.02855	50.03233	0	0	
$f_9(D = 30)$	Best	17.25501	13.36351	3.108206	0.065034	7.61E−06	6.48E−14	8.88E−16	1
	Worst	19.51932	18.09905	4.42218	0.266024	0.00078	6.72E−13	8.88E−16	
	Mean	18.84672	15.4624	3.64468	0.105065	0.000288	2.96E−13	8.88E−16	
	Std	0.677517	1.397258	0.363982	0.05465	0.000251	1.96E−13	0	
$f_{10}(D = 30)$	Best	150.6383	26.94152	0.221077	0.083807	5.91E−11	0	0	1
	Worst	379.3019	51.63634	0.477681	0.413484	0.05561	0	0	
	Mean	234.455	40.77981	0.342794	0.259248	0.01015	0	0	
	Std	62.87933	7.21818	0.063625	0.09981	0.01719	0	0	
$f_{11}(D = 30)$	Best	91,881,878	539,182.1	1.065289	1.658644	0.00034	0.072805	2.099679	2
	Worst	4.89E+08	5,552,265	3.22571	3.798615	2.360053	0.89254	2.797765	
	Mean	2.33E+08	2,678,210	2.053933	2.8477	1.033139	0.242317	2.472082	
	Std	1.22E+08	1,553,562	0.623148	0.567587	0.696304	0.191449	0.219079	

Table 7 Simulation results for test functions $f_i, i = 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23$

Benchmark functions	Result	Method							Rank
		ABC	CS	FPA	GWO	CGWO	SOS	CSOS	
$f_{12}(D=2)$	Best	0.998004	0.998004	12.67051	2.982105	15.10873	0.998004	**0.998004**	1
	Worst	2.317285	1.992261	12.67051	17.37441	455.8065	0.998004	0.998004	
	Mean	1.460413	1.150072	12.67051	9.508932	170.931	0.998004	0.998004	
	Std	0.506746	0.346879	5.31E−12	4.50628	166.9777	3.46E−16	1.42E−16	
$f_{13}(D=4)$	Best	0.001671	0.000732	0.000595	0.000389	0.000341	**0.0003076**	0.00030786	4
	Worst	0.018808	0.001709	0.001455	0.058957	0.037553	0.0012232	0.00171104	
	Mean	0.006427	0.001172	0.000871	0.006501	0.004337	0.0004081	0.00055943	
	Std	0.005313	0.000284	0.000276	0.015494	0.009319	0.0002357	0.00035062	
$f_{14}(D=2)$	Best	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 0.91277	− 1.03163	− **1.0316285**	1
	Worst	− 1.0312	− 1.03163	− 1.03031	− 1.02867	49.79596	− 1.03163	− 1.0316285	
	Mean	− 1.03159	− 1.03163	− 1.03145	− 1.03143	4.97676	− 1.03163	− 1.0316285	
	Std	0.00011	8.92E−08	0.000419	0.000765	12.66872	6.94E−14	8.3925E−17	
$f_{15}(D=2)$	Best	− 0.99904	− 0.99879	− 0.99992	− 1	− 0.93443	− 1	− **1**	1
	Worst	− 0.93598	− 0.93625	− 0.93624	− 0.93625	− 0.2593	− 1	− 1	
	Mean	− 0.95743	− 0.97592	− 0.9749	− 0.97355	− 0.60434	− 1	− 1	
	Std	0.024924	0.02598	0.02323	0.031732	0.198253	9.86E−15	0	
$f_{16}(D=2)$	Best	3.037584	3.000004	3.000001	3.000004	5.002017	3	**3**	1
	Worst	7.38301	3.001557	3.037702	84.00425	620.1465	3	3	
	Mean	3.818471	3.000254	3.005169	13.80229	132.4067	3	3	
	Std	1.282759	0.000485	0.010416	28.50153	158.7296	2.61E−15	9.19E−16	
$f_{17}(D=3)$	Best	− 3.86278	− 3.86278	− **3.86254**	− 3.86258	− 3.84917	− 3.86278	− 3.86278	2
	Worst	− 3.7238	− 3.86271	− 3.85275	− 3.85116	− 2.77296	− 3.86278	− 3.86278	
	Mean	− 3.84959	− 3.86277	− 3.86016	− 3.85942	− 3.52968	− 3.86278	− 3.86278	
	Std	0.035433	1.75E−05	0.003149	0.003232	0.364976	1.61E−15	1.73E−15	

Table 7 continued

Benchmark functions	Result	Method							Rank
		ABC	CS	FPA	GWO	CGWO	SOS	CSOS	
$f_{18}(D=6)$	Best	− **3.31972**	− 3.31826	− 3.12518	− 3.32165	− 3.32075	− 3.322	− 3.322	4
	Worst	− 3.20391	− 3.26892	− 1.82236	− 3.10994	− 3.06445	− 3.2031	− 3.2031	
	Mean	− 3.28162	− 3.3049	− 2.67372	− 3.23963	− 3.22413	− 3.29091	− 3.25031	
	Std	0.037323	0.014152	0.369375	0.092581	0.080269	0.053398	0.059857	
$f_{19}(D=4)$	Best	− 10.0617	− 9.91272	− 5.05115	− 10.1364	− 10.1532	− 10.1532	− **10.1532**	6
	Worst	− 5.39997	− 6.23033	− 4.87126	− 2.67205	− 2.63028	− 5.0552	− 2.68286	
	Mean	− 8.28654	− 8.77829	− 4.99705	− 8.09533	− 5.30961	− 8.48695	− 8.60906	
	Std	1.401545	0.917113	0.067809	3.013862	3.599475	2.282408	3.073116	
$f_{20}(D=4)$	Best	− 10.3891	− 10.3276	− 5.07467	− 10.3959	− 10.4028	− 10.4029	− **10.4029**	5
	Worst	− 3.83007	− 7.68927	− 4.9	− 2.74802	− 1.3118	− 3.7243	− 3.72366	
	Mean	− 7.69548	− 9.46173	− 5.03274	− 9.31918	− 4.63885	− 8.0357	− 9.06713	
	Std	2.234255	0.741857	0.04781	2.667534	3.62176	2.82989	2.765398	
$f_{21}(D=4)$	Best	− 10.524	− 10.5126	− 5.12362	− 10.5204	− 10.5364	− 10.5364	− **10.5364**	6
	Worst	− 3.67345	− 7.38929	− 4.74366	− 3.83299	− 2.42173	− 5.12848	− 2.87114	
	Mean	− 7.72775	− 9.65801	− 5.05572	− 10.0246	− 7.40201	− 9.73689	− 6.89818	
	Std	2.163838	0.778978	0.096681	1.713958	3.975401	1.895258	3.530932	
$f_{22}(D=2)$	Best	− 0.99999	− 1	− 0.01278	− 1	− 0.80377	− 1	− **1**	1
	Worst	− 0.95856	− 0.99998	− 0.01278	− 0.99983	− 1.3E−06	− 1	− 1	
	Mean	− 0.99177	− 1	− 0.01278	− 0.99997	− 0.16455	− 1	− 1	
	Std	0.011218	4.89E−06	0	4.24E−05	0.223295	0	0	
$f_{23}(D=2)$	Best	− 0.99028	− 0.99795	− 1	− 0.99028	− 0.96249	− 1	− **1**	1
	Worst	− 0.91673	− 0.98941	− 0.99916	− 0.99028	− 0.52626	− 0.99028	− 1	
	Mean	− 0.96463	− 0.9912	− 0.9999	− 0.99028	− 0.64955	− 0.9979	− 1	
	Std	0.027735	0.002732	0.000229	1.03E−07	0.117548	0.003418	0	
Count of algorithm found global minimum		0	1	1	2	2	10	14	

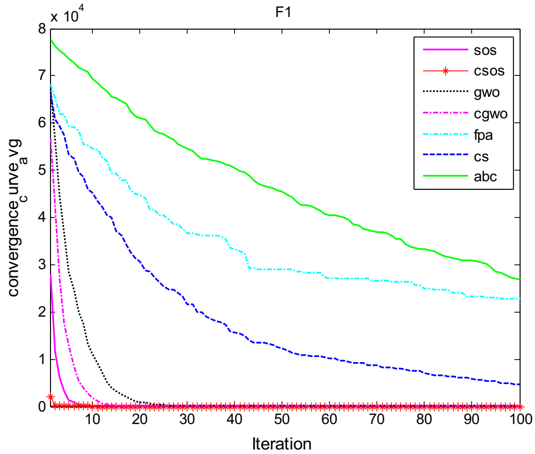


Fig. 2 $D = 30$, evolution curves of fitness value for f_{01}

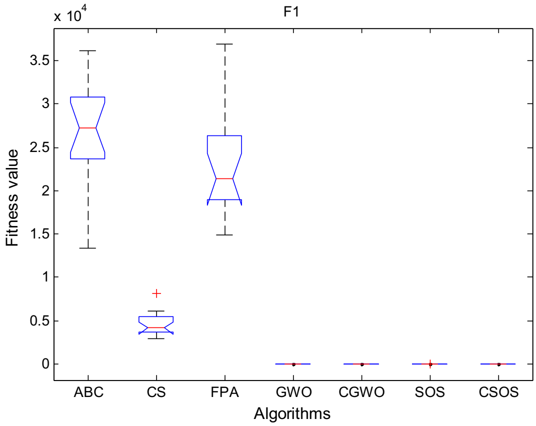


Fig. 3 $D = 30$, ANOVA test of global minimum for f_{01}

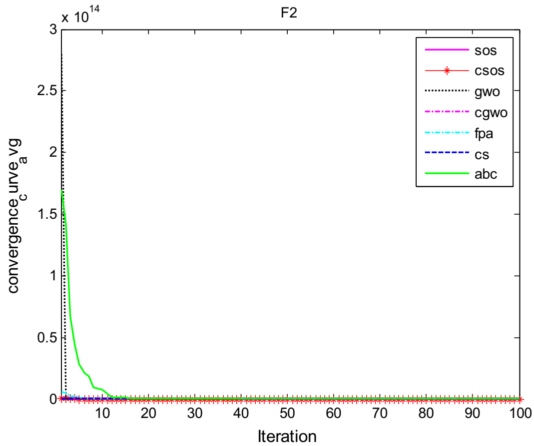


Fig. 4 $D = 30$, evolution curves of fitness value for f_{02}

Fig. 5 $D = 30$, ANOVA test of global minimum for f_{02}

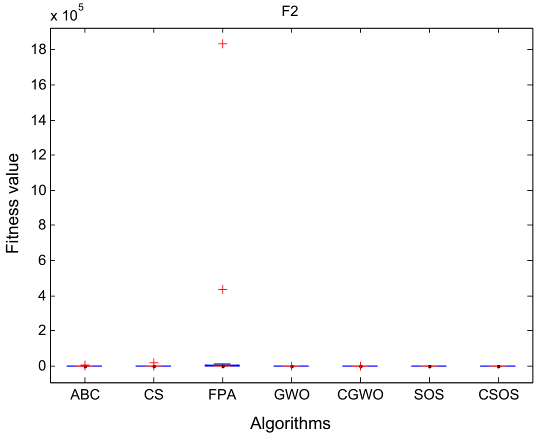


Fig. 6 $D = 30$, evolution curves of fitness value for f_{03}

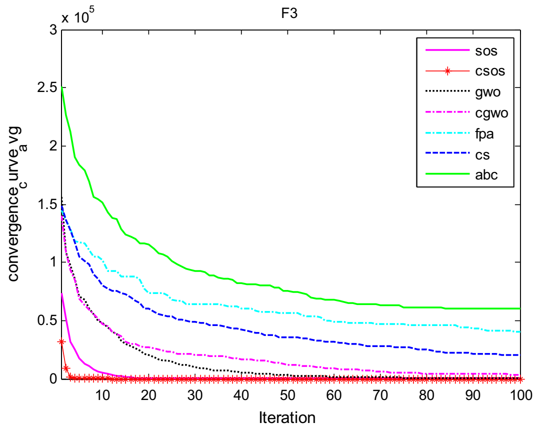


Fig. 7 $D = 30$, ANOVA test of global minimum for f_{03}

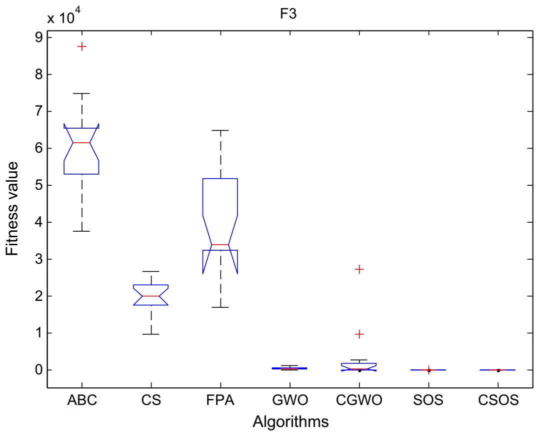


Fig. 8 $D = 30$, evolution curves of fitness value for f_{04}

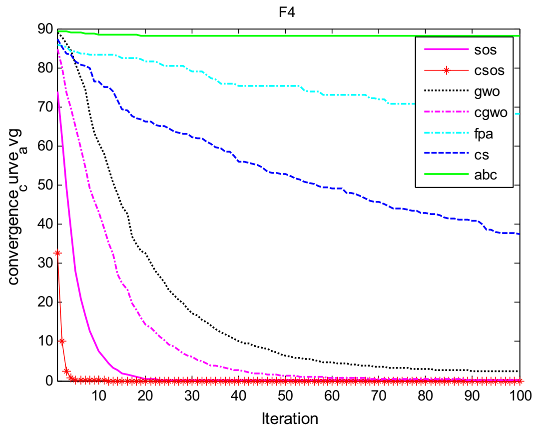


Fig. 9 $D = 30$, ANOVA test of global minimum for f_{04}

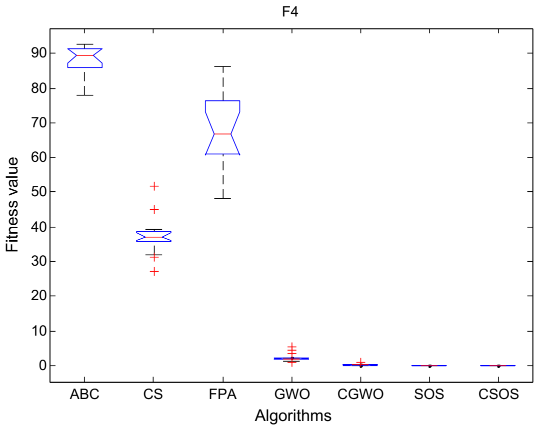
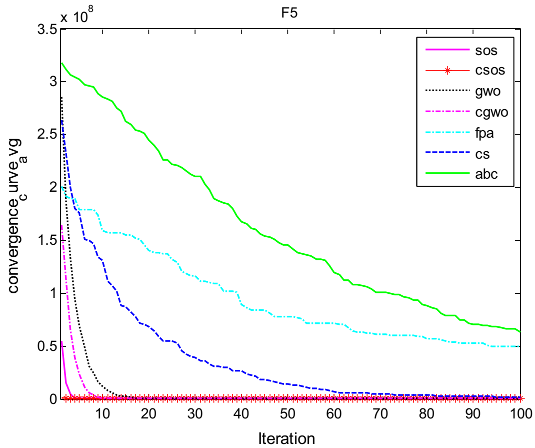


Fig. 10 $D = 30$, evolution curves of fitness value for f_{05}



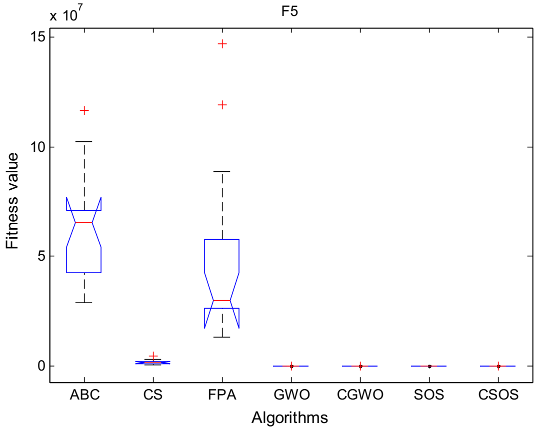


Fig. 11 $D = 30$, ANOVA test of global minimum for f_{05}

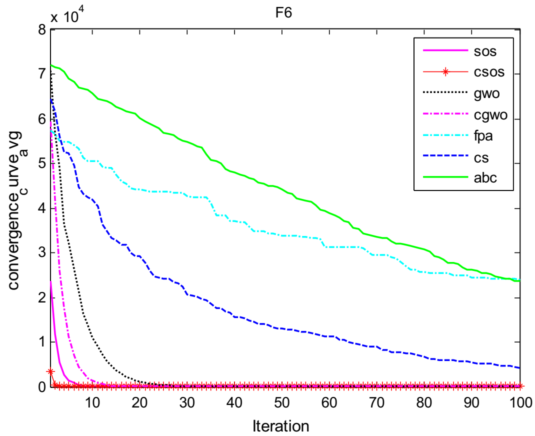


Fig. 12 $D = 30$, evolution curves of fitness value for f_{06}

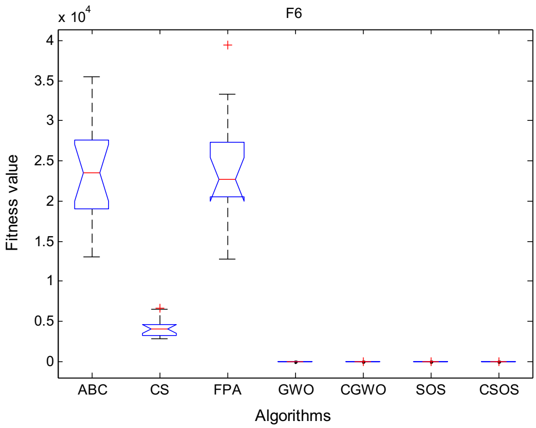


Fig. 13 $D = 30$, ANOVA test of global minimum for f_{06}

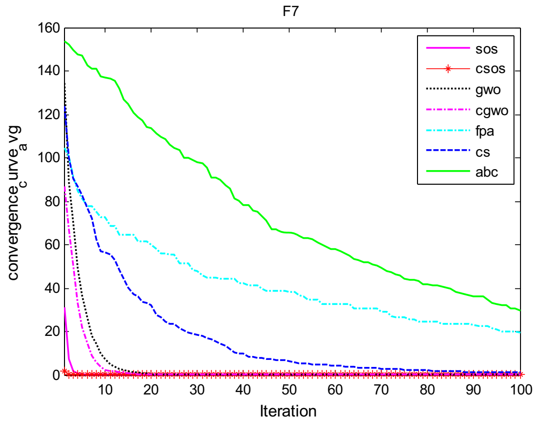


Fig. 14 $D = 30$, evolution curves of fitness value for f_{07}

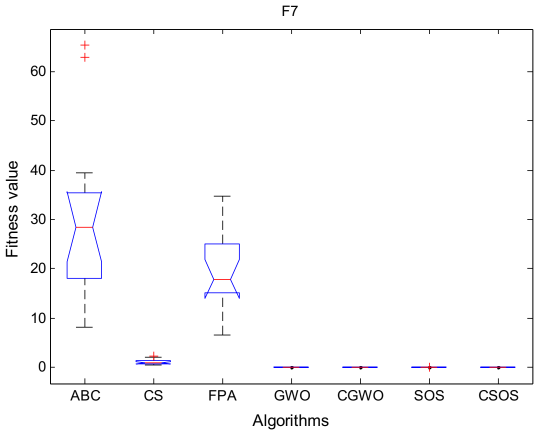


Fig. 15 $D = 30$, ANOVA test of global minimum for f_{07}

the figure $f_1 - f_7$ in the CSOS std. map flat, indicating that CSOS relative to other algorithms have a stronger robustness. From the convergence diagram can also be seen, CSOS in the convergence speed and accuracy is relatively fast, only in the f_5, f_6 convergence accuracy slightly worse than the SOS.

Similarly, according to the test results obtained in Table 6, the CSOS algorithm finds the theoretical optimal value zero of the multimodal benchmark functions f_8, f_{10}. This shows that compared with other algorithms, CSOS has a stronger ability to find the minimum. According to the mean value and the variance, we can see that it has high robustness in high-dimensional unimodal functions f_8, f_{10}. For function f_9, it can be seen from the optimal value and the average value in the test result that the minimum value found by CSOS is better than other algorithms. In addition, the standard deviation of CSOS is the least, which indicates that it has more stability than other algorithms. For the f_{11}, CSOS in the search accuracy on the poor, but the variance is smaller, higher stability. Figures 16, 17, 18, 19, 20, 21, 22 and 23 shows CSOS and other algorithm convergence and the anova tests of the global minimum plots. It can be seen that in addition to f_{11}, CSOS has higher convergence accuracy and stronger robustness in $f_8 - f_{10}$.

According to the test results in Table 7, it can be seen that CSOS has found the theoretical minimum in f_{12}, f_{14}, f_{15}, f_{16}, f_{19}, f_{20}, f_{21}, f_{22}, f_{23}. Meanwhile, in the functions

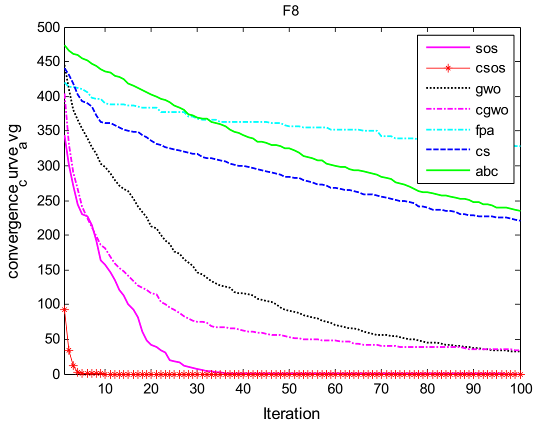


Fig. 16 $D = 30$, evolution curves of fitness value for f_{08}

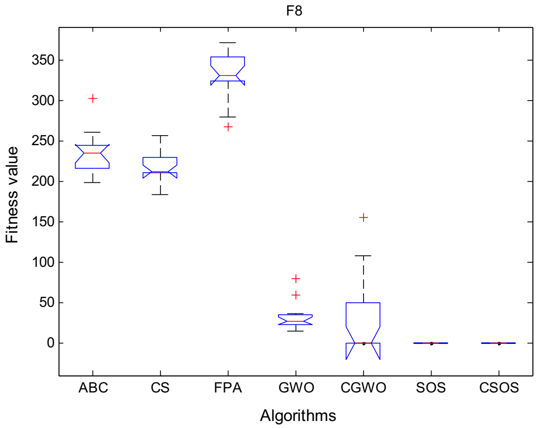


Fig. 17 $D = 30$, ANOVA test of global minimum for f_{08}

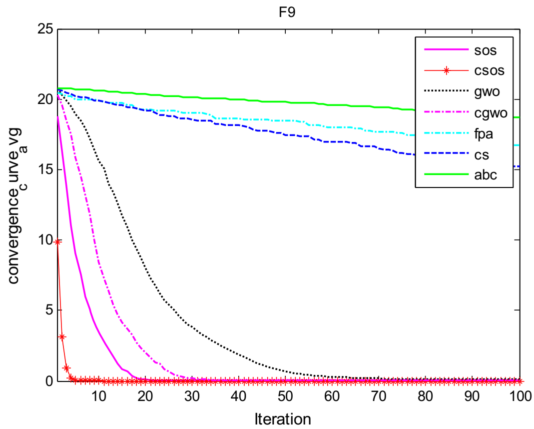


Fig. 18 $D = 30$, evolution curves of fitness value for f_{09}

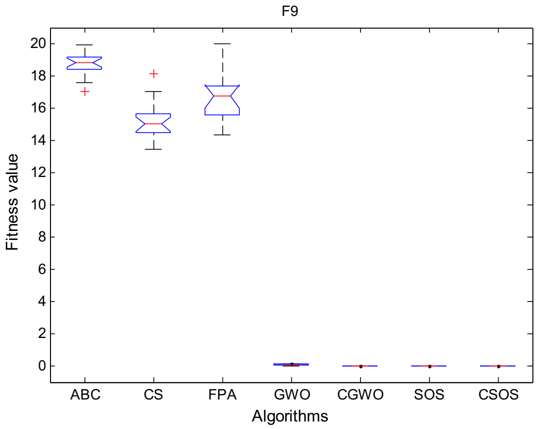


Fig. 19 $D = 30$, ANOVA test of global minimum for f_{09}

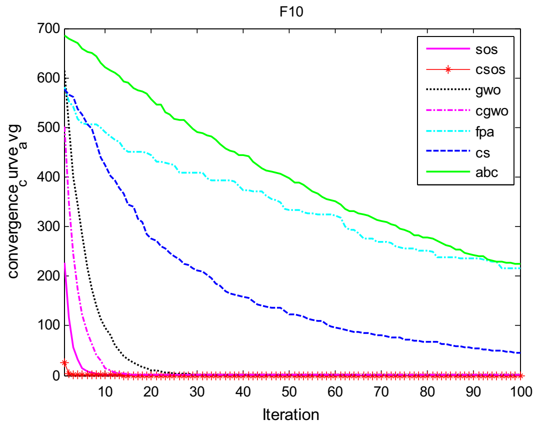


Fig. 20 $D = 30$, evolution curves of fitness value for f_{10}

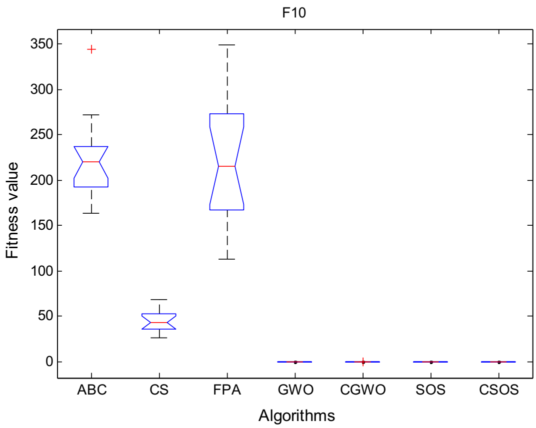


Fig. 21 $D = 30$, ANOVA test of global minimum for f_{10}

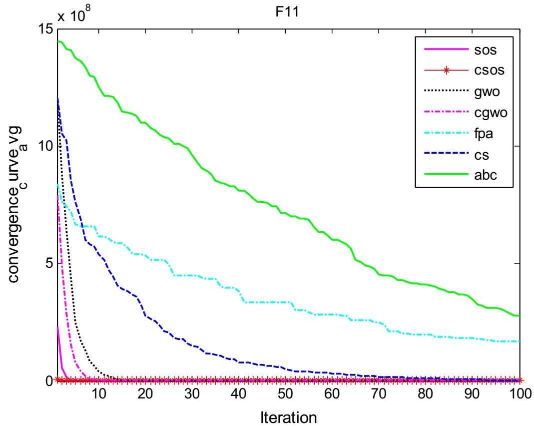


Fig. 22 $D = 30$, evolution curves of fitness value for f_{11}

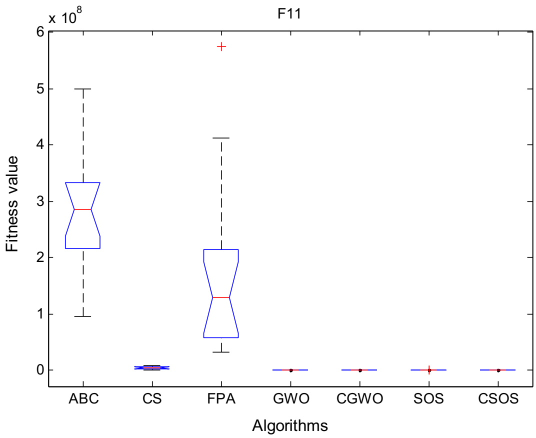


Fig. 23 $D = 30$, ANOVA test of global minimum for f_{11}

f_{12}, f_{14}, f_{15}, f_{16}, f_{22}, f_{23}, CSOS has a smaller standard deviation than other algorithms, which indicates that the CSOS has a stronger stability. For the function f_{13}, f_{17}, f_{18}, we can find that the optimal fitness value and standard deviation of CSOS are worse than other algorithms. For the function f_{21}, although the CSOS variance is the worst, but can be seen from the Table 7 CSOS find the global minimum, which shows that the f_{21} for the CSOS convergence accuracy but poor stability. Figures 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46 and 47 shows CSOS and other algorithm convergence and the anova tests of the global minimum plots. From the above data, it is easy to find that CSOS is also very competitive in terms of precision and robustness for fixed-dimension multimodal benchmark functions.

4.3 *p*-Values of the Wilcoxon rank-sum test

In this paper, the Wilcoxon rank-sum test [23,24] is used to verify the relationship between the CSOS algorithm and several other algorithms. The test to $p = 0.05$ as the standard, the test results are shown in Table 8.

In Table 8, data with p-values greater than 0.05 are indicated by bold and underlined. CSOS vs. ABC, CSOS vs. GWO and CSOS vs. CS have two values greater than 0.05 in

Fig. 24 $D = 2$, evolution curves of fitness value for f_{12}

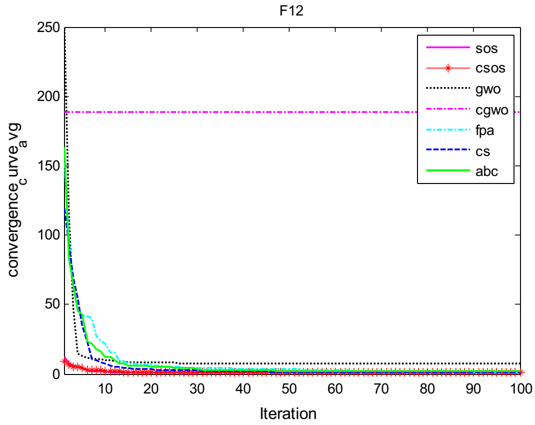


Fig. 25 $D = 2$, ANOVA test of global minimum for f_{12}

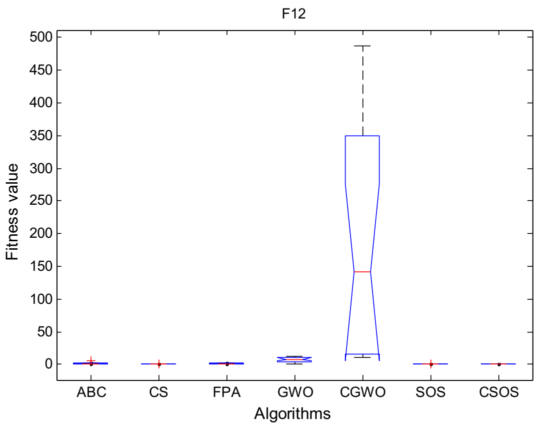


Fig. 26 $D = 4$, evolution curves of fitness value for f_{13}

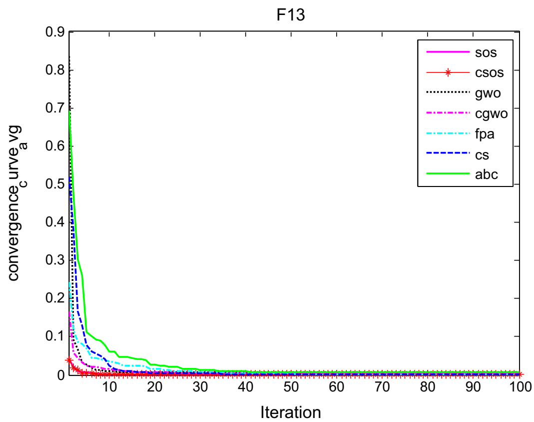


Fig. 27 $D = 4$, ANOVA test of global minimum for f_{13}

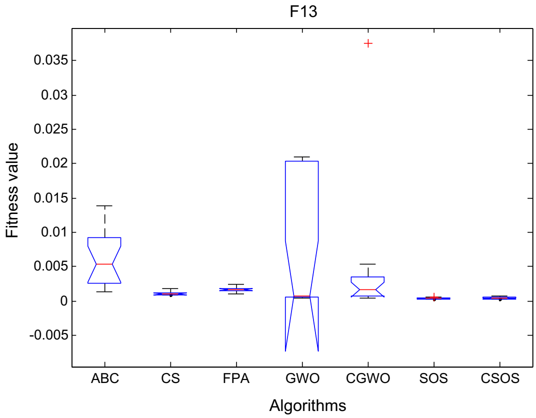


Fig. 28 $D = 2$, evolution curves of fitness value for f_{14}

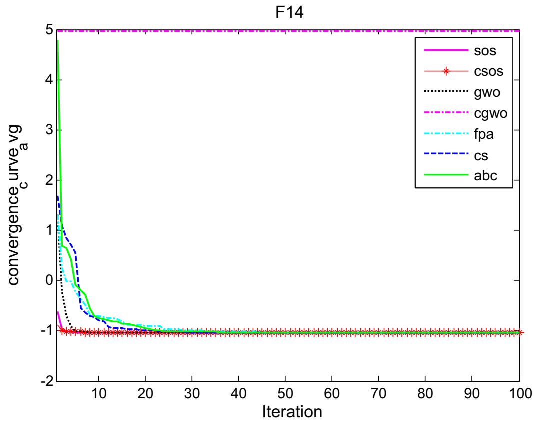
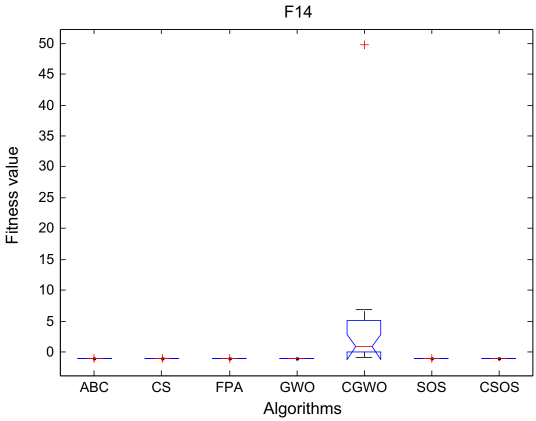


Fig. 29 $D = 2$, ANOVA test of global minimum for f_{14}



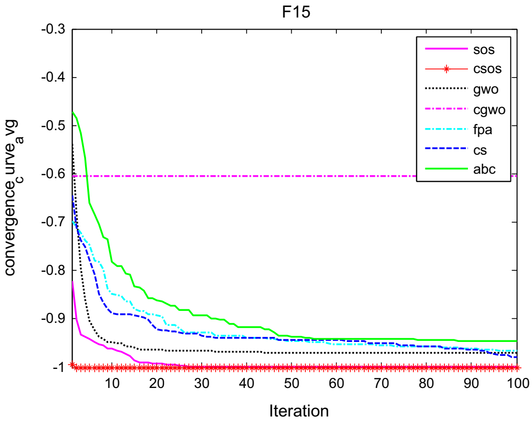


Fig. 30 $D = 2$, evolution curves of fitness value for f_{15}

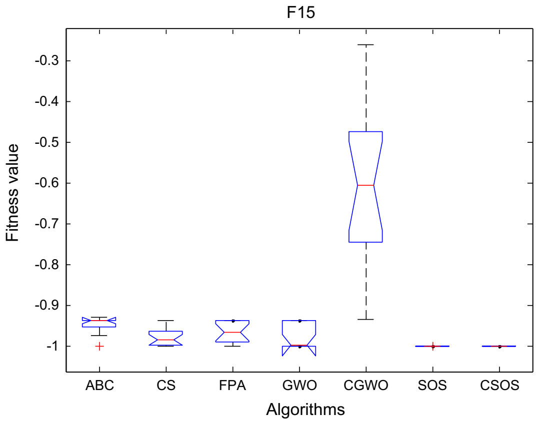


Fig. 31 $D = 2$, ANOVA test of global minimum for f_{15}

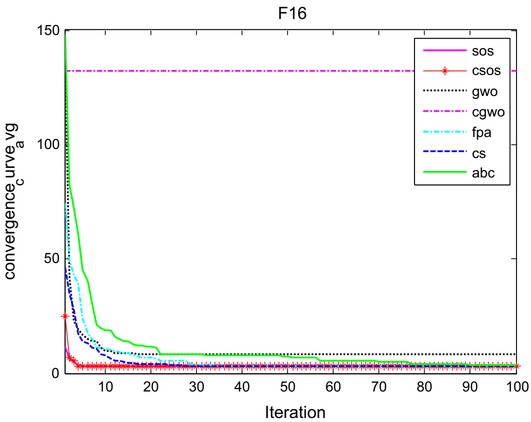


Fig. 32 $D = 2$, evolution curves of fitness value for f_{16}

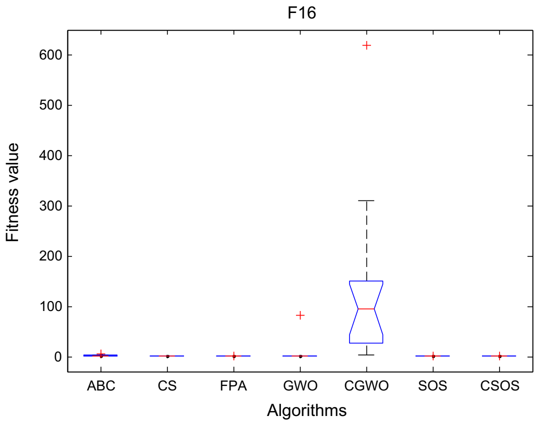


Fig. 33 $D = 2$, ANOVA test of global minimum for f_{16}

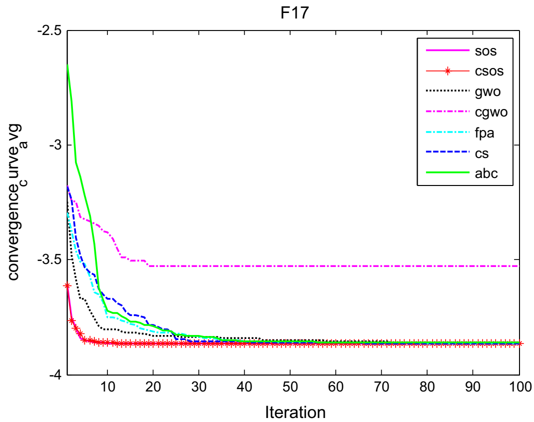


Fig. 34 $D = 3$, evolution curves of fitness value for f_{17}

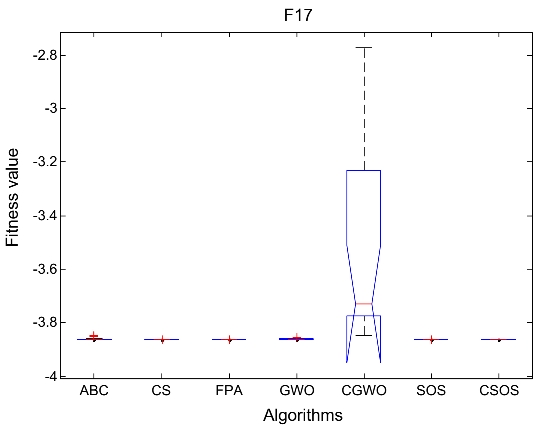


Fig. 35 $D = 3$, ANOVA test of global minimum for f_{17}

Fig. 36 $D = 6$, evolution curves of fitness value for f_{18}

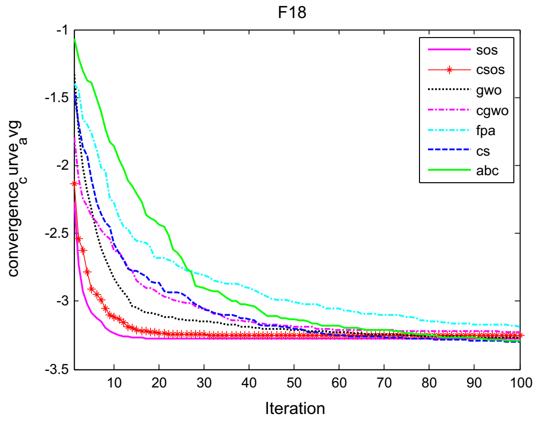


Fig. 37 $D = 6$, ANOVA test of global minimum for f_{18}

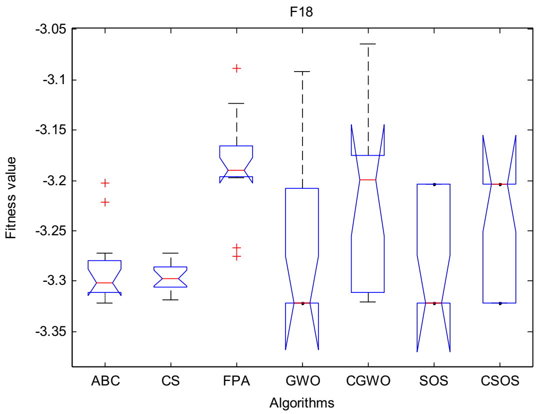
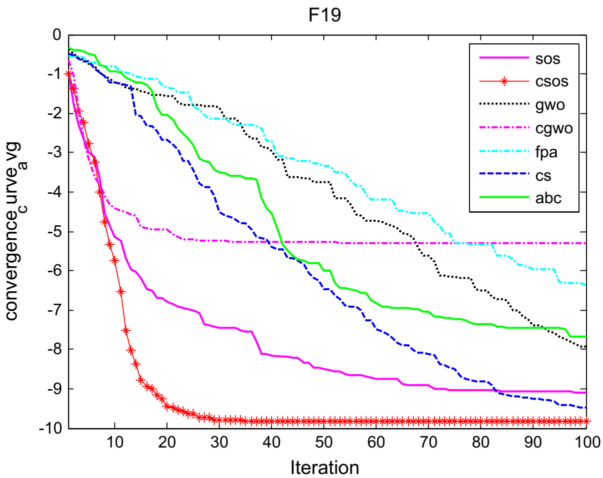


Fig. 38 $D = 4$, evolution curves of fitness value for f_{19}



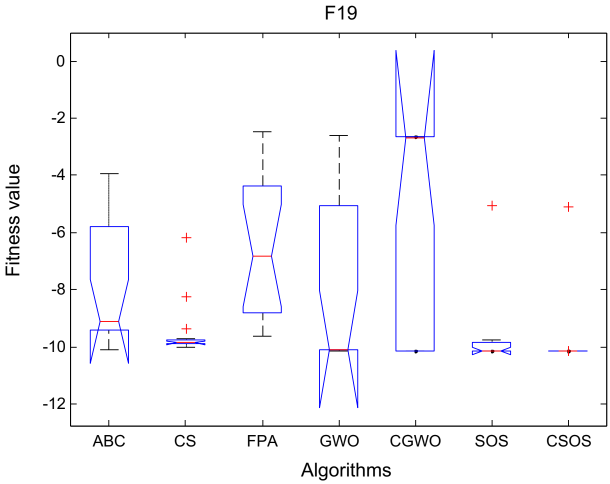
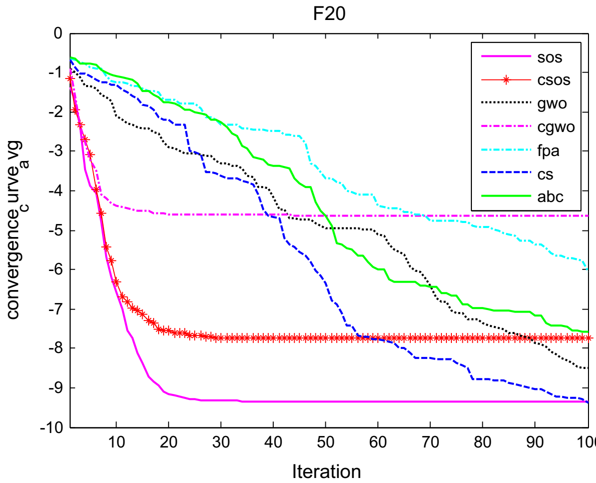


Fig. 39 $D = 4$, ANOVA test of global minimum for f_{19}

Fig. 40 $D = 4$, evolution curves of fitness value for f_{20}



each set of contrast data. CSOS vs. FPA, CSOS vs. CGWO have one value greater than 0.05 in each set of contrast data. We compare CSOS and SOS, there are three values greater than 0.05. All other values were less than 0.05. Therefore, there are significant differences between CSOS and other algorithms. The experimental data are not obtained by accident.

4.4 CSOS for engineering optimization problem

In order to verify the effectiveness of CSOS for complex problems, this paper chooses two engineering examples of cantilever beam design optimization problem [25] and welding beam design optimization problem [26] to validate this project.

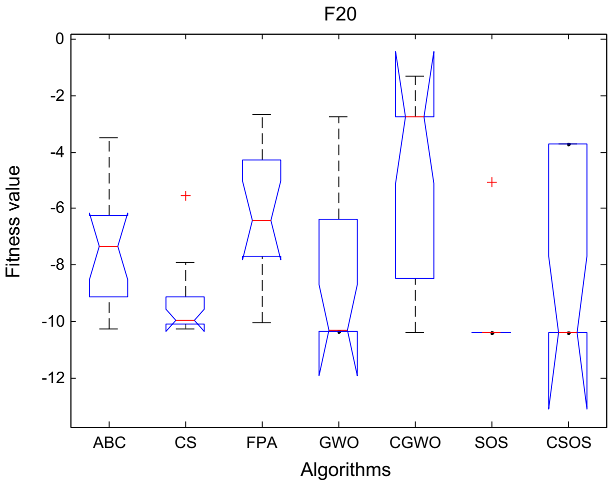


Fig. 41 $D = 4$, ANOVA test of global minimum for f_{20}

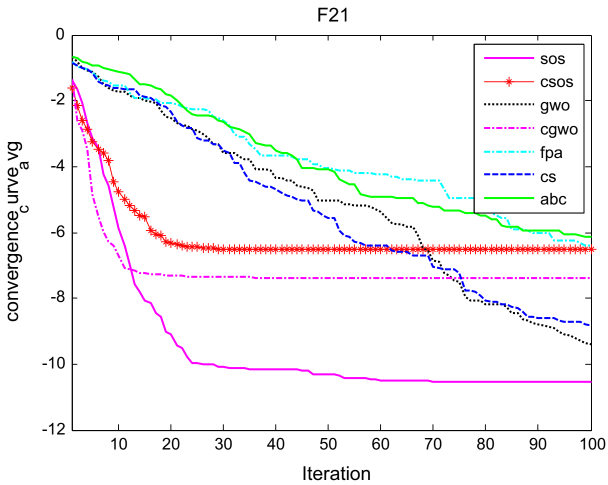


Fig. 42 $D = 4$, evolution curves of fitness value for f_{21}

4.4.1 Cantilever beam design problem

Cantilever structure shown in Fig. 48 [25], which is composed of five square hollow structure, each component with a variable. It can be seen from the figure that there are a downward force on the point 6 and a fixed support at the point 1. The objective is to minimize the weight of the beam. The problem formulation is as follows:

$$\text{Minimize } f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5);$$
$$\text{Subject to } g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1;$$

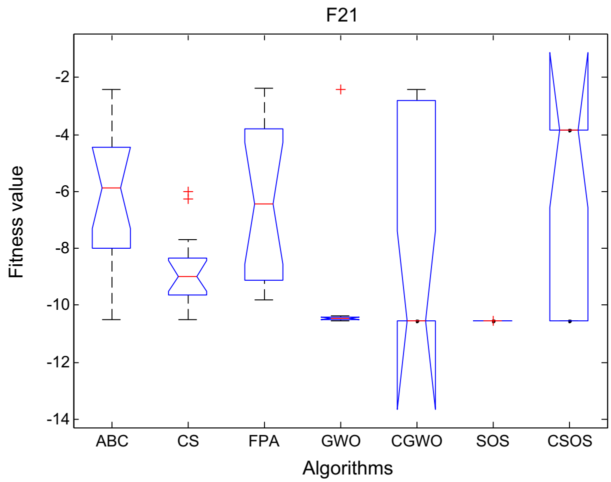


Fig. 43 $D = 4$, ANOVA test of global minimum for f_{21}

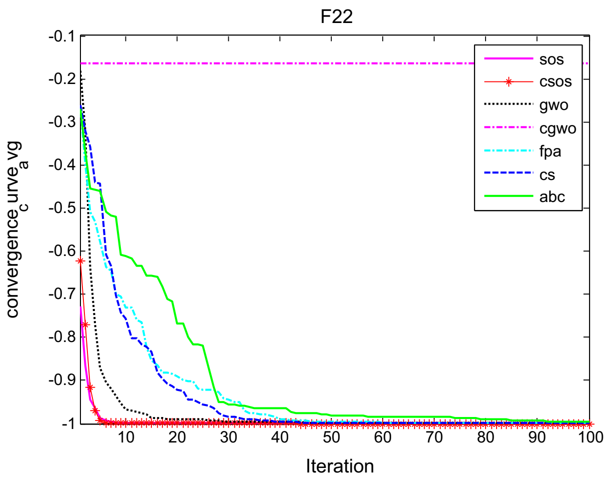


Fig. 44 $D = 2$, evolution curves of fitness value for f_{22}

In this paper, the CSOS algorithm was tested with Method of Moving Asymptotes (MMA) [26], Generalized Convex Approximation (GCA_I) [26], GCA_II [26], CS [27], and Symbiotic Organisms Search (SOS) [27] in 20 independent experiments. The test results are shown in Table 9.

It can be seen from the data in Table 9 that CSOS can find a better optimal value than other algorithms. This shows the superiority of the CSOS algorithm in solving the cantilever problem.

4.4.2 Welded beam design problem

The purpose of the welded beam design problem is to obtain the minimum fabricating cost. The structural design of the welded beam is shown in Fig. 49 [26]. The constraints are as follows: shear stress (τ), bending stress in the beam (θ), end deflection of the beam (δ),

Fig. 45 $D = 2$, ANOVA test of global minimum for f_{22}

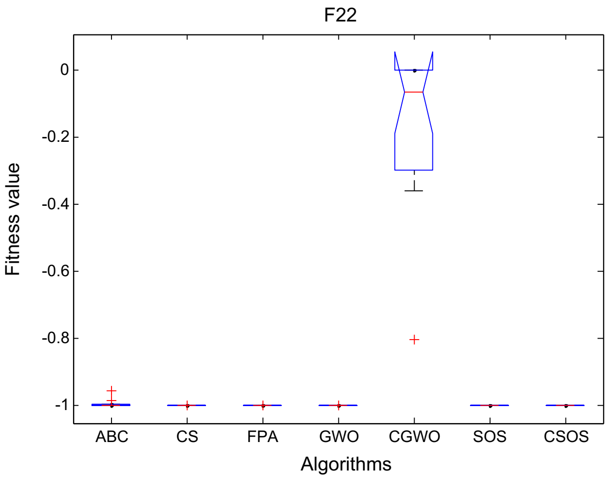


Fig. 46 $D = 2$, evolution curves of fitness value for f_{23}

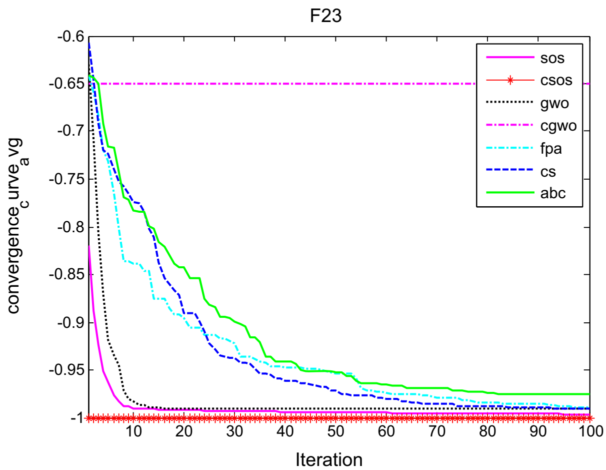


Fig. 47 $D = 2$, ANOVA test of global minimum for f_{23}

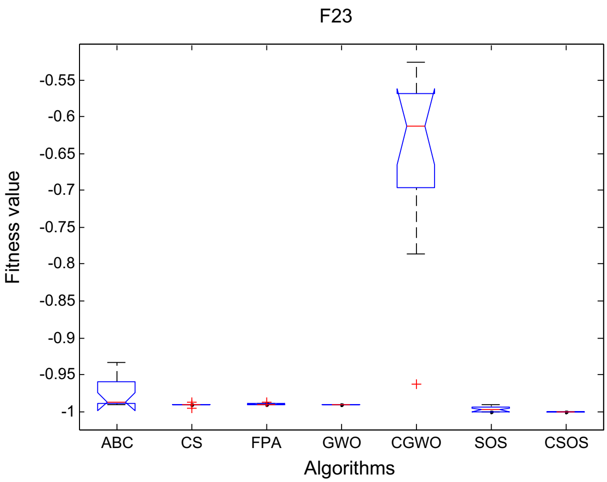


Table 8 *p* Values of the Wilcoxon rank-sum test results

Functions	CSOS vs ABC	CSOS vs CS	CSOS vs FPA	CSOS vs GWO	CSOSvs CGWO	CSOS versus SOS
f_{01}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07
f_{02}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07
f_{03}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07
f_{04}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07
f_{05}	3.38E−06	3.38E−06	3.38E−06	3.38E−06	0.022471	3.38E−06
f_{06}	3.38E−06	3.38E−06	3.68E−06	0.000779	3.39E−06	3.38E−06
f_{07}	3.39E−06	3.39E−06	3.39E−06	3.39E−06	3.39E−06	3.39E−06
f_{08}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	N/A
f_{09}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.82E−07
f_{10}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	N/A
f_{11}	3.39E−06	3.39E−06	0.022531	0.012822	1.94E−05	3.39E−06
f_{12}	1.88E−06	1.88E−06	1.88E−06	1.88E−06	1.85E−06	0.000171
f_{13}	4.14E−06	5.74E−05	0.000576	0.000906	0.000223	0.018067
f_{14}	1.26E−06	1.26E−06	1.26E−06	1.26E−06	1.85E−06	0.001104
f_{15}	6.87E−07	6.87E−07	6.87E−07	0.003625	6.87E−07	0.350648
f_{16}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	9.58E−07	6.54E−07
f_{17}	1.85E−06	1.85E−06	1.85E−06	1.85E−06	2.36E−06	0.14952
f_{18}	0.276036	0.294928	2.7E−06	0.180038	0.037298	0.048264
f_{19}	0.008073	0.008073	0.004844	0.008073	3.66E−06	0.028743
f_{20}	0.004377	0.004377	0.004377	0.001904	0.000214	0.005285
f_{21}	1	0.771229	0.771229	1	0.197055	0.382459
f_{22}	6.87E−07	6.87E−07	8.27E−08	6.87E−07	6.87E−07	N/A
f_{23}	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07	6.87E−07

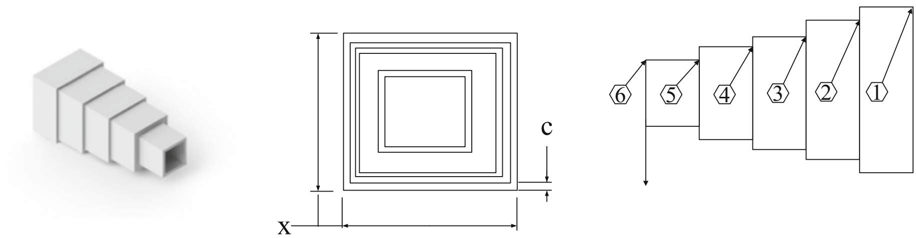


Fig. 48 Cantilever beam design problem

buckling load on the bar (P_c), and side constraints. The four design variables associated with this problem are as follows:

- Thickness of the weld (h)
- Length of the welded joint (l)
- Width of the bar (t)
- Thickness of the bar (b)

The formula involved in the design of welded beam is as follows:

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

Table 9 Comparison results for cantilever design problem

Algorithm	Optimal values for variables					Optimal value
	x_1	x_2	x_3	x_4	x_5	
CSOS	6.01579434	5.3093581	4.4943115	3.501481	2.1527151	1.3399564
MMA [26]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_I [26]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II [26]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [27]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [27]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996

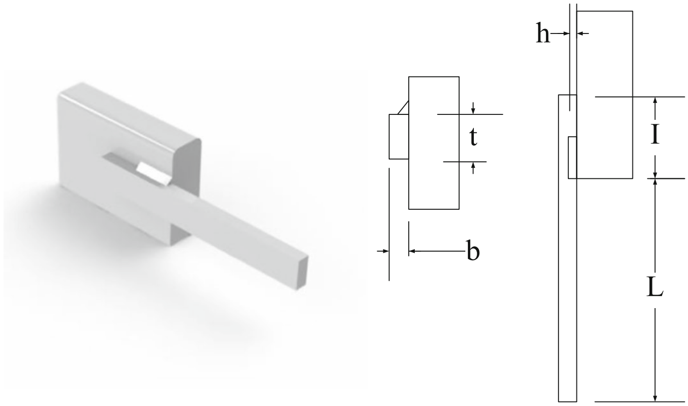


Fig. 49 Design parameters of the welded beam design problem

$$
\begin{aligned}
\text{Subject to } g_1(x) &= \tau(x) - \tau_{\max} \le 0,\\
g_2(x) &= \sigma(x) - \sigma_{\max} \le 0,\\
g_3(x) &= x_3 - x_4 \le 0,\\
g_4(x) &= 0.125 - x_1 \le 0,\\
g_5 &= \delta(x) - 0.25 \le 0,\\
g_6 &= P - P_c(x) \le 0,\\
g_7 &= 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \le 0;
\end{aligned}
$$

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

$$
\begin{aligned}
\text{Where} \quad \tau(x) &= \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2};\\
\tau_1 &= \frac{P}{\sqrt{2}x_1x_2};\\
\tau_2 &= \frac{MR}{J};\\
M &= P\left(L + \frac{x_2}{2}\right);
\end{aligned}
$$

Table 10 Comparison results for cantilever design problem

Algorithm	Optimal values for variables				Optimal value
	h	l	t	b	
CSOS	0.2057296	3.253120	9.03662391	0.2057296398	1.695247
GWO [9]	0.205676	3.478377	9.03681	0.205778	1.72624
GSA [9]	0.182129	3.856979	10.0000	0.202376	1.87995
CPSO [9]	0.202369	3.544214	9.048210	0.205723	1.72802
GA (Coello) [28]	N/A	N/A	N/A	N/A	1.8245
GA (Deb) [29]	N/A	N/A	N/A	N/A	2.3800
GA (Deb) [30]	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem) [31]	0.2442	6.2231	8.2915	0.2443	2.3807
Random [32]	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex [32]	0.2792	5.6256	7.7512	0.2796	2.5307
David [32]	0.2434	6.2552	8.2915	0.2444	2.3841
Approx [32]	0.2444	6.2189	8.2915	0.2444	2.3815

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

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

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

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

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

The test was carried out independently 20 times; the test results shown in Table 10. The CSOS and GWO [9], GSA [9], CPSO [9], GA (Coello) [28], GA (Deb) [29], GA (Deb) [30], HS (Lee and Geem) [31], Random [32], Simplex [32], David [32] and APPROX [32] of the 20 independent experiments to verify the validity of CSOS for welding beam problem, the results shown in Table 10.

Compared with other algorithms, CSOS found a higher solution in the design of the welded beam, and the relevant parameters are $h = 0.2057296$, $l = 3.253120$, $t = 9.03662391$, $b = 0.2057296398$. This experiment shows the effectiveness of CSOS in the welding beam problem.

4.5 Result analysis

Simulation experiments have been done in Sects. 4.2 and 4.3. In Sect. 4.2, 23 standard benchmark functions were used to verify all aspects of CSOS performance. The experimental data are shown in Tables 5, 6 and 7. Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46 and 47 shows the evaluation curves of fitness values, anova test of global minimum. According to the test data and figure can be seen, CSOS has a stronger ability to find the global minimum and better stability. In the Sect. 4.3, the test of CSOS Wilcoxon with other algorithms; the result is not accidental. In Sect. 4.4, two engineering examples are selected to verify the validity of the CSOS. The experimental results are shown in Tables 9 and 10. The results show that CSOS in engineering problems also have high accuracy and stability.

5 Conclusions

In this paper, the idea of complex-valued coding is incorporated into the symbiotic organisms search (SOS) algorithm, and a novel complex-valued encoding symbiotic organisms search (CSOS) algorithm is proposed. CSOS takes advantage of the feature of complex-valued encoding, that is, the two-dimensional coding space maps one-dimensional coding space, real and imaginary parts are updated separately, and each biological individual has inherent parallelism, which increases the population diversity and enhances the ability of the algorithm to find the global minimum. CSOS extends the application range of the symbiotic organisms search algorithm from a real number range to a complex number range. From the results of the 23 standard benchmark functions tests in this paper, CSOS has better optimization precision and stability than other algorithms. In future studies, it is recommended that CSOS be applied to more real-world engineering problems and solve some NP- hard problems in literature.

Acknowledgements This work is supported by National Science Foundation of China under Grant Nos. 61463007, 61563008. Project of Guangxi University for Nationalities Science Foundation under Grant No. 2016GXNSFAA380264.

References

1. Socha K, Dorigo M (2008) Ant colony optimization for continuous domains. Eur J Oper Res 185(3):1155–1173
2. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997(11):341–359
3. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Perth, Australia, vol IV, pp 1942–1948
4. Yang XS (2012) Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation. Lecture notes in computer science, vol 7445, pp 240–249
5. Yang XS, Deb S (2009) Cuckoo search via levy flights. In: World congress on nature and biologically inspired computing (NaBIC 2009). IEEE Publication, USA, pp 210–214
6. Yang XS (2013) Multiobjective firefly algorithm for continuous optimization. Eng Comput 29(2):175–184
7. Kaveh A, Zolghadr A (2011) Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm. Asian J Civ Eng 12:487–509
8. Yang X (2010) A new metaheuristic bat-inspired algorithm. In: Gonzalez JR, Pelta DA, Cruz C (eds) Nature inspired cooperative strategies for optimization. Springer, Berlin, pp 65–74
9. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
10. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
11. Abdullahi M, Ngadi A Md, Abdulhamid SM (2016) Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Gen Comput Syst 56:640–650

12. Secui DC (2016) A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point effects. Energy 113:366–384
13. Prasad D, Mukherjee V (2016) A novel symbiotic organisms search algorithm for optimal power flow of power system with FACTS devices. Int J Eng Sci Technol 19:79–89
14. Das B, Mukherjee V, Das D (2016) DG placement in radial distribution network by symbiotic organisms search algorithm for real power loss minimization. Appl Soft Comput 49:920–936
15. Casasent D, Natarajan S (1995) A classifier neural network with complex-valued weights and square-law nonlinearities. Neural Netw 8:989–998
16. Chen D-B, Li H-J, Li Z (2009) Particle swarm optimization based on complex-valued encoding and application in function optimization. Comput Eng Appl 45:59–61
17. Zheng Z, Zhang Y, Qiu Y (2003) Genetic algorithm based on complex-valued encoding. Control Theory Appl 20(1):97–100
18. Panda A, Pani S (2016) A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Appl Soft Comput 46:344–360
19. Tang K, Yao X, Suganthan PN, MacNish C, Chen Y-P, Chen C-M, Yang Z (2007) Benchmark Functions for the CEC'2008 special session and competition on large scale global optimization. University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, China, Technical Report. http://nical.ustc.edu.cn/cec08ss.php
20. Hansen N, Auger A, Finck S, Ros R (2009) Real-parameter black-box optimization benchmarking 2009 experimental setup. Institute National de Recherche en Informatique et en Automatique (INRIA), Rapports de Recherche RR-6828, 20 Mar 2009
21. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
22. Luo Q, Zhang S, Li Z, Zhou Y (2015) A novel complex-valued encoding Grey Wolf optimization algorithm. Algorithms 9(1):4
23. Wilcoxon F (1944) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
24. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15:617. https://doi.org/10.1007/s10732-008-9080-4
25. Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39(5):829–846
26. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–27
27. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
28. Carlos A, Coello C (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Syst 17:319–46
29. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:31–338
30. Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015
31. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933
32. Ragsdell K, Phillips D (1976) Optimal design of a class of welded structures using geometric programming. ASME J Eng Ind 98:1021–1026

Fahui Miao received the B.S. degree in Computer Science from Henan Institute of Technology, Henan, China, in 2011, He is currently a M.S. with Guangxi University for Nationalities. His research interests include computation intelligence, intelligence information processing.

Yongquan Zhou received the M.S. degree in Computer Science from Lanzhou University, Lanzhou, China, in 1993, and the Ph.D. degree in Computation Intelligence from Xidian University, Xi'an, China, in 2006. He is currently a Professor with Guangxi University for Nationalities. His research interests include computation intelligence, neural networks, intelligence information processing, big data.

Qifang Luo received the M.S. degree in Computer Science from Guangxi University in 2005. Her is currently a Professor with Guangxi University for Nationalities. Her research interests are in the areas of computation intelligence and intelligence information systems.