A complex-valued encoding seeker optimization algorithm for constrained engineering problems

SHAOMI DUAN [1,2], HUILONG LUO [1], HAIPENG LIU [2]
[1] The Faculty of Civil Engineering and Mechanics, Kunming University of Science and Technology, Kunming 650500, CHINA.
[2] The Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, CHINA.

Abstract: - This article comes up with a complex-valued encoding seeker optimization algorithm (CSOA) base on the multi-chain method for the constrained engineering optimization problems. The complex value encoding and a multi-link strategy are leaded by the seeker optimization algorithm (SOA). The complex value encoding method is an influential global optimization strategy, and the multi-link is an enhanced local search strategy. These strategies enhance the individuals' diversity and avert fall into the local optimum. This article chose fifteen benchmark functions, four PID control parameter models, and six constrained engineering problems to test. According to the experimental results, the CSOA algorithm can be used in the benchmark functions, PID control parameters optimization, and optimization constrained engineering problems. Compared to particle swarm optimization (PSO), simulated annealing base on genetic algorithm (SA_GA), gravitational search algorithm (GSA), sine cosine algorithm (SCA), multi-verse optimizer (MVO), and seeker optimization algorithm (SOA), the optimization ability and robustness of CSOA are better.

Key-Words: - complex-valued encoding; multi-link; seeker optimization algorithm; functions optimization; PID control parameters optimization problems; constrained engineering optimization problems

Received: January 18, 2021. Revised: June 30, 2021. Accepted: July 14, 2021. Published: July 25, 2021.

1 Introduction

Recently, the heuristic algorithm has received a lot of attention. Such algorithms create random methods for many optimization problems. As we all know, since the No Free Lunch (NFL) theorem, no one optimization solution can optimize overall questions [1]. Therefore, researchers to pose new algorithms or enhance the current algorithms to deal with optimization problems. The current algorithms are genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], simulated annealing (SA) [4], harmony search (HS) [5], gravitational search algorithm (GSA) [6], Moth-flame optimization (MFO) [7], sine cosine algorithm (SCA) [8], multi-verse optimizer (MVO) [9], seeker optimization algorithm (SOA) [10].

However, some optimization algorithms are still not very successful in many optimization problems, such as low optimization precision, premature, only local optimal solution, slow convergence speed, insufficient robustness, etc. To better overcome the issues of common optimization precision, premature, only local optimal solution, slow convergence rate, poor robustness， the complex-valued encoding heuristic algorithms have been proposed according to the characteristics of some algorithms, which enhance the algorithms. These complex-valued encoding intelligent optimization algorithms have been proved to be feasible optimization algorithms and have been used in many practical engineering. For instance, the plural encoding dragonfly algorithm optimizes the power systems [11]. A gray wolf optimization based on plural encoding optimizes the filter model [12]. The plural encoding satin bowerbird optimization algorithm solves benchmark functions [13]. The plural encoding driven optimization optimizes the 0-1 knapsack problem [14]. The plural encoding symbiotic organisms search algorithm is proposed for overall optimization [15]. The plural encoding flower pollination algorithm optimizes constrained engineering optimization problems [16]. A comprehensive survey is offered for plural encoding metaheuristic optimization algorithm [17].

Dai et al. propose the SOA algorithm in 2006 [18]; the goal is to mimic the behavior of seekers and the way they exchange information, solving practical application optimization problems. The recent decade, the SOA algorithm has been used in many fields, such as unconstrained optimization problems [19], optimal reactive power dispatch [20], a challenging set of benchmark problems [21], the design of digital filter [22], optimizing parameters of artificial neural networks [23], the optimizing model and structures of fuel cell [24], the novel human group optimizer algorithm [25], and several practical applications [26].

However, in the initial stage of dealing with optimization problems, SOA converges faster than others; When all individuals are near to the best individual for solving the optimization problem, the individuals will lose diversity and fall into premature.

To overcome the shortcomings of SOA, in this article, complex number coding and a multi-link strategy are used to enhance global optimization and local search. The CSOA has been tested on nine unimodal functions, six multimodal functions, four PID control parameters optimizations, and six engineering optimizations taken from the literature. In comparison with the PSO, SA_GA, GSA, SCA, MVO, and SOA, the CSOA can find better values to solving the questions, the precision and robustness of the CSOA algorithm are better. The complex value encoding and multi-link methods enhance the diversity of the individuals and avert premature. Our CSOA algorithm overcomes the premature of SOA. The advantages of CSOA are summed up as follows.

A CSOA algorithm is raised to enhance the precision and robustness in the optimization process. With the multi-chain strategy, in complex-valued coding, the real part, imaginary part, and real number are used as parallel individual variables to solve the objective function problem. The multi-chain strategy can improve the diversity of individuals and boost partial scouting.

Introduce the stochastic multi-chain strategy. According to the initial solution generation rule of complex number coding, the real part, imaginary part, and real number are randomly generated as parallel individual variables to solve the objective function. It can avert premature.

The rest of the article structure is as follows. Part 2 presentations the SOA. Part 3 describes the CSOA. Part 4 shows algorithms optimization experiments and results in analyses. At last, Part 5 gives some conclusions.

2 Basic SOA Algorithm

The SOA algorithm carries out in-depth research on human search behavior. It considers optimization as search for an optimal solution by a search team in search space, taking search team as population and the site of the searcher as task method. Using "experience gradient" to determine the search direction, using uncertain reasoning to resolve the search step measurement, through the scout direction and search step size to complete the searcher's position in the search interspace update, to attain the optimization of the solution.

2.1 Search direction

The forward orientation of search is defined by the experience gradient obtained from the individuals' movement and the evaluation of other individuals' search historical position. The egoistic direction $\vec{f}_{i,\mathrm{e}}(t)$, altruistic direction $\vec{f}_{i,\mathrm{a}}(t)$ and preemptive direction $\vec{f}_{i,\mathrm{p}}(t)$ of the ith individual in any dimension can be obtained, i.e.

$$\vec{f}_{i,\mathrm{e}}(t)=\vec{p}_{i,\mathrm{best}}-\vec{x}_i(t) \tag{1}$$

$$\vec{f}_{i,a}(t)=\vec{g}_{i,\mathrm{best}}-\vec{x}_i(t) \tag{2}$$

$$\vec{f}_{i,\mathrm{p}}(t)=\vec{x}_i(t_1)-\vec{x}_i(t_2) \tag{3}$$

The searcher uses the method of random weighted average to obtain the search orientation, i.e.

$$\vec{f}_i(t)=\mathrm{sign}(\omega\vec{f}_{i,\mathrm{p}}(t)+\varphi_1\vec{f}_{i,\mathrm{e}}(t)+\varphi_2\vec{f}_{i,\mathrm{a}}(t)) \tag{4}$$

Where: $t_1,t_2\in\{t,t-1,t-2\}$, $\vec{x}_i(t_1)$ and $\vec{x}_i(t_2)$ are the best advantages of $\{\vec{x}_i(t\text{-}2),\ \vec{x}_i(t\text{-}1),\ \vec{x}_i(t)\}$ separately; $g_{i,\mathrm{best}}$ is the historical optimal location in the neighborhood where the ith search factor is located; $p_{i,\mathrm{best}}$ is the optimal locality from the ith search factor to the current locality; and φ_1 are φ_2 random numbers in [0,1]. ω is the weight of inertia.

2.2 Search step size

The SOA algorithm refer to the reasoning approximation ability of the fuzzy system. The SOA algorithm through the computer language describe some of the human natural language which can simulate human intelligence reasoning search behavior. If the algorithm expresses in a simple fuzzy rule, adapts to the best approximation of the objective optimization problems. The greater the search step length is more significant, on the contrary, the smaller the fitness, step length and the corresponding smaller. The gaussian distribution function is adopted to describe the search step measurement, namely

$$\mu(\alpha)=e^{-\frac{\alpha^2}{2\delta^2}} \tag{5}$$

Where, α and δ are parameters of membership function.

According to equation (5), the probability of the output variable exceeding [-3δ,3δ] is less than 0.0111. Therefore, $\mu_{\min}=0.0111$. Under normal circumstances, the optimal position of an individual has $\mu_{\max}=1.0$ and the worst place is 0.0111.

However, to accelerate the convergence speed and get the optimal individual to have uncertain step size, $\mu_{\max}$ is set as 0.9 in this paper. Select the following function as the fuzzy variable with a "small" target function value:

$$\mu_i = \mu_{\max} - \frac{s - I_i}{s - I}(\mu_{\max} - \mu_{\min}), i = 1, 2, \cdots, s \tag{6}$$

$$\mu_{ij} = \text{rand}(\mu_i, 1), j = 1, 2, \cdots, D \tag{7}$$

Where: μ_{ij} is determined by equations (6) and (7), and I_i is the count of the sequence $x_i(t)$ of the current individuals arranged from high to low by function value. And the function $rand(\mu_i, 1)$ is the real number in any partition $[\mu_i, 1]$.

It can be seen from equation (6) that it simulates the random search behavior of human beings. Step measurement of j-dimensional search interspace is determined by equation (8):

$$\alpha_{\text{ij}} = \delta_{ij} - \sqrt{-\ln(\mu_{ij})} \tag{8}$$

Where, δ_{ij} is a parameter of Gaussian distribution function, which is defined by equation (9):

$$\delta_{ij} = \omega x * \text{abs}(\vec{x}_{\min} - \vec{x}_{\max}) \tag{9}$$

Where, ω is the weight of inertia. As the evolutionary algebra increases, ω decreases linearly from 0.9 to 0.1. $\vec{x}_{\min}$ and $\vec{x}_{\max}$ are respectively the variate of the minimum value and maximum value of the function.

2.3 Individual location updates

After obtaining the scout direction and scout step measurement of individual, the location update is represented by (10):

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) f_{ij}(t), i = 1, 2, \cdots, s; j = 1, 2, \cdots, D \tag{10}$$

Where, *i* is the ith searcher individual, *j* represents the individual dimension; $f_{\text{ij}}(t)$ and $\alpha_{\text{ij}}(t)$ respectively represent the searcher's search direction and search step size at time t, $x_{ij}(t)$ and $x_{ij}(t+1)$ respectively represent the searchers' site at time *t* and (*t*+1).

3 CSOA Algorithm

The chromosomes of complex organisms are double-stranded or multi-stranded construction. Since the two-dimensional structure of complex-valued encoding [14, 27-29], it is represented as a two-chain. An individual consists of double chains of an identical length, and the real and imaginary parts of a plural can be used to represent a chromosome pair. The two-body framework enhances the variety of individuals and makes the algorithm have better searching and calculation capacity.

3.1 Initial population generation

In the light of the variable interval [A_k, B_k], k=1, 2, ⋯,2*M*-1, 2*M*, modules ρ_k, phase angles θ_k and plural are produced [30] as following formula (11)-(13).

$$\rho_k = \frac{A_k, B_k}{2} * \text{rand} \in [0, \frac{A_k, B_k}{2}], k = 1, 2, \cdots, 2M-1, 2M \tag{11}$$

$$\theta_k = 4\pi(\text{rand-0.5}) \in [-2\pi, 2\pi], k = 1, 2, \cdots, 2M-1, 2M \tag{12}$$

$$X_{Rk} + \text{i}X_{Ik} = \rho_k(\cos\theta_k + i\sin\theta_k), k = 1, 2, \cdots, 2M-1, 2M \tag{13}$$

The real parts X_{Rk} and imaginary parts X_{Ik} update by section 3.2.

3.2 Individual location updates

The real part is updated by formula (14):

$$x_R(t+1) = x_R(t) + \alpha_R(t) f_R(t) \tag{14}$$

Where, α_R represents the scout direction of the real parts, and f_R is the scout step measurement of the real parts. X_R represents the location of the real number parts, *t* represents time *t*, and (*t*+1) represents time (t+1).

The imaginary part is updated by formula (15):

$$x_I(t+1) = x_I(t) + \alpha_I(t) f_I(t) \tag{15}$$

Where, α_I represents the scout direction of the imaginary part, f_I is the scout step measurement of the imaginary part, X_I represents the location of the imaginary number part, t represents time *t*, and (*t*+1) represents time (*t*+1).

3.3 Fitness evaluation method

When calculating fitness values using SOA algorithms, we convert plural to real numbers. The formula is as follows:

(1) Take the plural mathematical module as real number:

$$\rho_{\text{k}} = \sqrt{X_{Rk}^2 + X_{Ik}^2}, k = 1, 2, \cdots, M-1, M \tag{16}$$

(2) Define the sign according to the phase angle:

$$X_{\text{k}} = \rho_{\text{k}} \,\text{sgn}(\sin(\frac{X_{Ik}}{\rho_{\text{k}}}))\frac{B_k + A_k}{2}, k = 1, 2, \cdots, M-1, M \tag{17}$$

X_k is the real number.

3.4 Multi-chain strategy

The multi-chain strategy includes taking the real and imaginary part of the plural as separate parallel solutions, and randomly generating parallel solutions according to the complex number coding law.

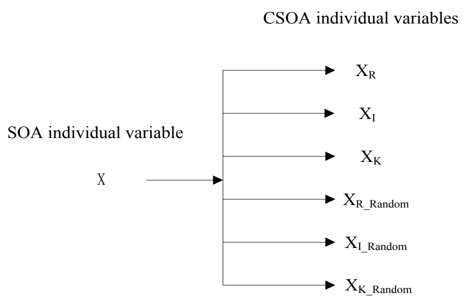


Fig. 1 Multi-chain strategy of CSOA

In this paper, the meaning of the multi-chain strategy is that a single individual variable in the original SOA algorithm is converted into six parallel individual parameters when the CSOA optimize a problem. In complex-valued coding, there are real part XR, imaginary part XI, and real number XK. In each iterative loop optimization, XR, XI and XK are adjusted to variables that meet the scope of X ($X_{min}=A_k$, $X_{max}=B_k$). XR, XI, and XK were taken as the relative optimal solution variables respectively to solve the objective function problem. Secondly, a group of variables that randomly generate X_{R_Random}, X_{I_Random}, and X_{K_Random} according to formulas (11), (12), (13) and meet the scope of X ($X_{min}=A_k$, $X_{max}=B_k$) should be added in each cycle optimization, and taken as the relative optimal solution variable to solve the objective function respectively. At the end individual of the single solution, the respective optimal solutions are saved, and the global optimal value is saved as the current optimal value after the comparison of each optimal solution. The optimal solution variables of the next generation of XR, XI, and XK are changed according to formulas (15), (16), and (17). The next generation optimal solution variables of X_{R_Random}, X_{I_Random}, and X_{K_Random} are generated randomly according to formulas (11), (12), and (13). In other words, a single individual variable X in the original SOA algorithm is converted to six individual variables X_R, X_I, X_K, X_{R_Random}, X_{I_Random}, and X_{K_Random} when solved by the CSOA algorithm, and this is shown in Fig. 1. So instead of solving for one main chain, we are solving for six parallel chains. CSOA algorithm uses a multi-chain strategy, which adds the variety of the individual enhance local scout, and averts premature convergence.

3.5 CSOA algorithm process

The complex-valued encoding seeker optimization algorithm (CSOA) is based on a multiple population evolution model, three populations evolved by SOA, and three other populations evolved from random generation. Individual groups use information-sharing mechanisms to realize coevolution. Algorithm 1 is the primary process of CSOA.

Algorithm 1: CSOA

1. $t \leftarrow 0$
2. **Initialization** Generate initial species group based on formula (11), (12), and (13).
3. Convert plural into real numbers based on the formulas (16) and (17).
4. Determine the range of $X_{R\_CSOA,G}$, $X_{I\_CSOA,G}$, and $X_{CSOA,G}$ to satisfy the range of X.
5. **Evaluate** each seeker. Computing the fitness.
6. While stopping condition is not satisfied
   6.1 Running process of the CSOA algorithm
   6.1.1 Renew the real parts by formula (14), $X_{R\_CSOA,G}$.
   6.1.2 Renew the imaginary parts based on formula (15), $X_{I\_CSOA,G}$.
   6.1.3 Convert plural into real number based on formula (16) and (17), $X_{CSOA,G}$.
   6.1.4 Determine the range of $X_{R\_CSOA,G}$, $X_{I\_CSOA,G}$, and $X_{CSOA,G}$ to satisfy the range of X.
   6.1.5 Scout strategy giving scout direction and scout range.
   6.1.6 Calculate the fitness $f(X_{R\_CSOA,G})$, $f(X_{I\_CSOA,G})$, $f(X_{CSOA,G})$.
   6.1.7 Identify the best solution $X_{CSOAbest,G}$
   $F_{CSOA,G}=\min[f(X_{R\_CSOA,G})\ \ f(X_{I\_CSOA,G})\ \ f(X_{CSOA,G})]$
   $X_{CSOAbest,G}=\min(F_{CSOA,G})$
   6.2 Random generation and calculation
   6.2.1 Generate Initial population according to formula (11), (12), and (13).
   6.2.2 Convert complex numbers into real numbers according to formulas (16) and (17).
   6.2.3 Determine the range of $X_{R\_Random,G}$, $X_{I\_Random,G}$, and $X_{Random,G}$ to satisfy the range of X.
   6.2.4 Calculate the fitness $f(X_{R\_Random,G})$, $f(X_{I\_Random,G})$, $f(X_{Random,G})$.
   6.2.5 Identify the best value $X_{Randombest,G}$
   $F_{Random,G}=\min[f(X_{R\_Random,G})\ \ f(X_{I\_Random,G})\ \ f(X_{Random,G})]$
   $X_{Randombest,G}=\min(F_{Random,G})$
   6.3 Confirm the global best value X_{best}
   If $f(X_{CSOAbest,G}) \le f(X_{Randombest,G})$
   $X_{best}=X_{CSOAbest,G}$
   else $X_{best}=X_{Randombest,G}$
   end if
7. $t=t+1$
8. if $t<T_{max}$, then jump to 3; Else Stop.

4 Experimental Results

4.1 Experimental setup

All algorithms used in the experiment in this paper were running under MATLAB R2016a. The computer is configured as Intel (R) Core (TM) i7-7500U CPU @2.7GHz 2.9 GHz processor with 8 GB of memory, the operating system is Windows 10.

4.2 Parameters Setting

In the subsection, the parameters setting of PSO [31], SA_GA [32], GSA [6], SCA [8], MVO [9], SOA [18] and CSOA algorithm are presented. According to the references [6] [8] [11] [18] [31] [32], we did a lot of practice tests and comparative studies about the parameters. The parameters of the seven algorithms depend on the real experience to take the right value. Tab. 1 lists the parameters in the test. To ensure the comparison is fair, the seven algorithms the population number is 30, and the evolutionary algebra is 1000. At the same time, for further ensuring the fairness of algorithm comparison and reducing the effect of randomness, the results of the seven algorithms after 30 independent runs were selected for comparison.

Tab. 1 The parameters set of algorithms

Algorithm	Parameters and Value
PSO [31]	Constant inertia: 0.9~0.4, The two acceleration coefficients: 1.4962.
SA_GA [32]	Select probability:0.6, Crossover probability:0.7, Mutation scale factor:0.05, Original temperature:100, Temperature reduction parameter:0.98.
GSA [6]	The gravitational constant: G0=100, alfa=20.
SCA [8]	The random numbers: r1=0~2, r2=0~2π, r3=0~2, r4=0~1.
MVO [9]	Probability of wormhole existence: WEP_Max =1, WEP_Min=0.2, travelling distance rate: TDR=0~1, the random numbers: r1=0~1, r2=0~1, r3=0~1.
SOA [18]	The membership degree value: MDV_Max =0.95, MDV_Min =0.0111, The inertia weight value: IWV_Max =0.9, IWV_Min=0.1.
CSOA	The membership degree value: MDV_Max =0.95, MDV_Min =0.0111, The inertia weight value: IWV_Max =0.9, IWV_Min=0.1.

4.3 Algorithm performance comparison in benchmark functions

Using fifteen benchmark functions [7,10,33-35] to test the capability of the CSOA algorithm, the functions have been widely used in the test. CSOA compares with the PSO, SA-GA, GSA, SCA, MVO, and SOA algorithms.

4.3.1 Algorithm performance comparison in unimodal benchmark functions

In this field, it is common to base the capability of algorithms on mathematic functions that are known to be globally optimal. Nine unimodal reference functions in the literature were used as the comparative test platform [7,10,33-35].

4.3.1.1 Unimodal benchmark functions

Tab. 2 has shown the unimodal functions in the experiment. Variables are set to one thousand.

Tab. 2 Description of unimodal functions

Functions	Range	Minimum
$f_1(x)=\sum_{i=1}^{n}x_i^2$	[-100,100]	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10]	0
$f_3(x)=\sum_{i=1}^{n}\sum_{j=1}^{i}x_j^2$	[-100,100]	0
$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	[-100,100]	0
$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	[-30,30]	0
$f_6(x)=\sum_{i=1}^{n}(\lvert x_i+0.5\rvert)^2$	[-100,100]	0
$f_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	[-1.28,1.28]	0
$f_8(x)=\sum_{i=1}^{n}ix_i^2$	[-10,10]	0
$f_9(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{(i+1)}$	[-1.28,1.28]	0

4.3.1.2 Results comparison of algorithms in unimodal benchmark functions

The mean values, standard deviation (Std.) values, best values, and best values rank between the algorithms of 30 all alone runs; the results of the functions $f1$-$f9$ are displayed in Tab. 3. The underline and boldface indicate that the optimal result is better.

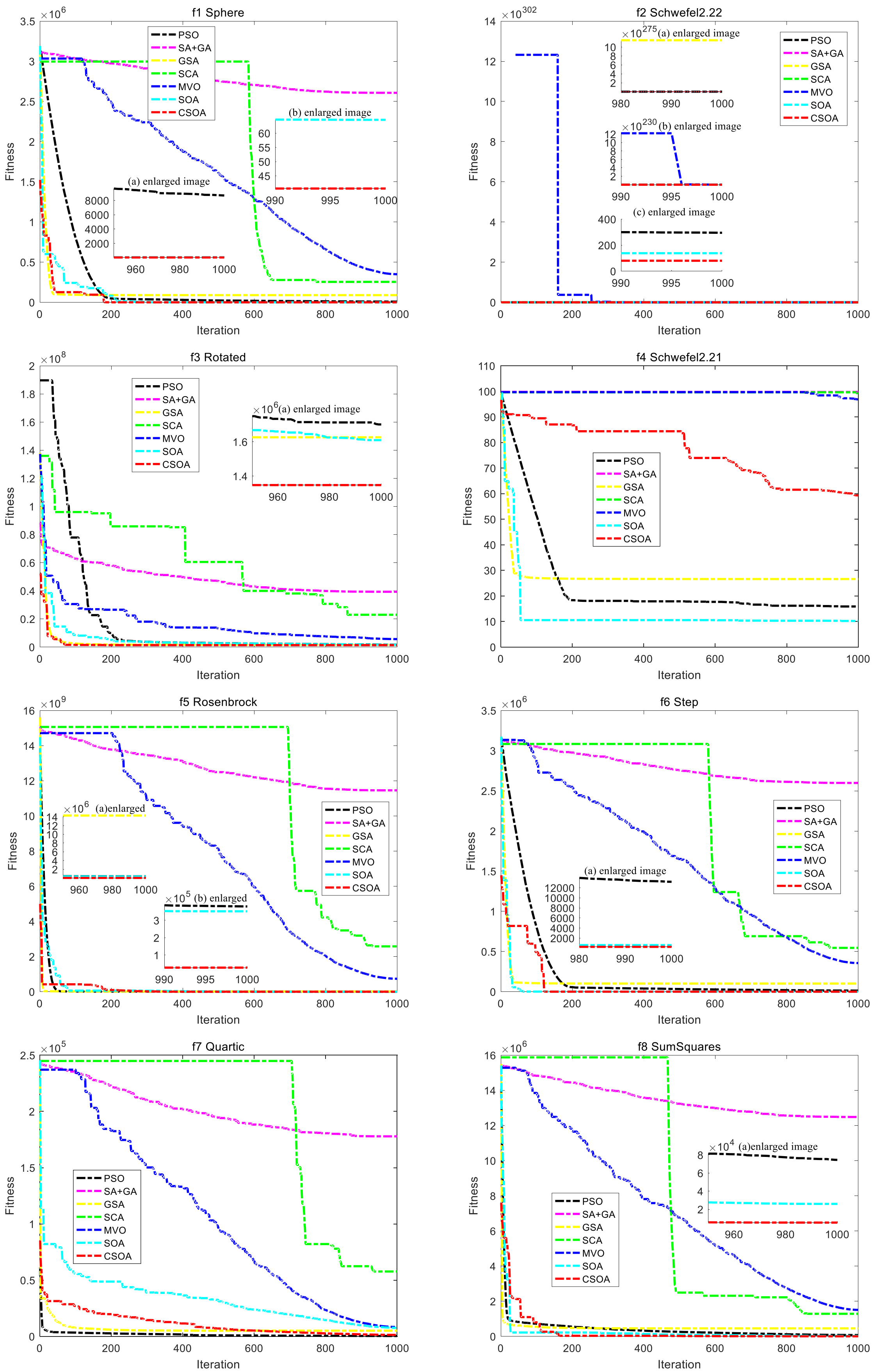
For the unimodal functions, based on Tab. 3, except $f4$, $f7$, and $f9$, the optimal value of the CSOA algorithm is better than others. To $f9$, the optimal value of CSOA has reached the theoretical best value, although the optimal fitness value of CSOA is worse than PSO and GSA, it the optimal fitness value result of CSOA to $f7$ function is only worse than PSO algorithm, the optimal fitness value result of CSOA to $f4$ is worse than PSO, GSA, and SOA algorithm. Except $f2$, $f3$, $f4$, $f7$, and $f9$, the standard deviation results of the CSOA algorithm are better than others. The standard deviation results of CSOA to $f9$ is only worse than PSO algorithm, the result of CSOA to $f7$ is worse than PSO, GSA, and SOA, the result of CSOA to $f4$ is worse than SCA, GSA, SA_GA, PSO, and MVO, the result of CSOA to $f3$ is worse than MVO and SOA, the result of CSOA to $f2$ is only worse than SOA. The standard deviation results of CSOA to $f2$ function of PSO, SA_GA, and SCA algorithm have no solution, the GSA and MVO algorithm have an infinite standard deviation. Except $f3$, $f4$, $f7$, and $f9$, the mean values of CSOA are better than others. To $f9$, the mean test results of CSOA has reached the theoretical best value, although the mean test result of CSOA is worse than PSO, the result of CSOA to $f4$ is worse than PSO, GSA, and SOA, the result of CSOA to $f3$ is only worse than GSA and SOA. According to the optimal fitness value mean rank and all rank results from Tab. 3, the CSOA has strong optimization ability and strong robustness to unimodal function.

Tab. 3 Performance comparison of algorithms for unimodal functions

Functions	Result	Algorithm PSO	SA_GA	GSA	SCA	MVO	SOA	CSOA
f_1 (D=1000)	Mean	9.7477e+3	2.5731e+6	9.7519e+4	3.5105e+5	3.5977e+5	1.1631e+3	**2.7677e+2**
	Std.	1.6106e+3	5.6740e+4	4.9089e+3	1.3646e+5	1.6121e+4	2.1241e+3	**1.1208e+3**
	Best	6.9495e+3	2.4536e+6	8.7230e+4	4.0211e+4	3.3044e+5	69.63801	**13.41123**
	Rank	3	7	5	4	6	2	**1**
f_2 (D=1000)	Mean	Inf	Inf	1.8010e+281	Inf	1.3436e+273	1.3642e+2	**84.4833**
	Std.	NaN	NaN	Inf	NaN	Inf	**5.69798**	6.8676
	Best	2.9025e+2	Inf	4.6083e+244	Inf	2.3145e+209	1.2392e+2	**74.4520**
	Rank	3	6	5	6	4	2	**1**
f_3 (D=1000)	Mean	3.3245e+6	3.3451e+7	1.9611e+6	2.4026e+7	6.1928e+6	**1.3029e+6**	2.0443e+6
	Std.	1.4068e+6	1.1343e+7	8.1529e+5	4.7320e+6	**4.3229e+5**	4.9041e+5	7.7353e+5
	Best	1.6748e+6	1.9532e+7	9.5585e+5	1.3982e+7	5.5268e+6	1.8272e+5	**1.7429e+4**
	Rank	4	7	3	6	5	2	**1**
f_4 (D=1000)	Mean	**18.3429**	99.5440	28.7444	99.5223	97.4126	28.6012	61.3865
	Std.	1.3182	0.1918	1.7962	**0.1313**	0.7405	10.4316	3.1861
	Best	16.3774	98.9588	25.8834	99.1965	95.6086	**2.3434**	53.9155
	Rank	2	6	3	7	5	**1**	4
f_5 (D=1000)	Mean	2.7249e+5	1.119e+10	1.6750e+7	3.2676e+9	6.7902e+8	2.2675e+5	**4.8206e+4**
	Std.	1.4792e+5	3.343e+8	1.7268e+6	7.0048e+8	7.3930e+7	1.2833e+5	**2.7505e+4**
	Best	1.2499e+5	1.057e+10	1.4932e+7	2.0841e+9	5.4843e+8	2.8880e+4	**1.1542e+4**
	Rank	3	7	4	6	5	2	**1**
f_6 (D=1000)	Mean	1.0284e+4	2.5635e+6	9.8733e+4	4.1315e+5	3.5579e5	1.3438e+3	**2.7896e+2**
	Std.	1.3817e+3	4.0795e+4	4.9212e+3	1.2787e+5	1.7687e+4	2.2562e+3	**66.6179**
	Best	6.1887e+3	2.4668e+6	9.0772e+4	1.6278e+5	3.1703e+5	2.5694e+2	**2.1193e+2**
	Rank	3	7	4	5	6	2	**1**
f_7 (D=1000)	Mean	**1.1474e+2**	1.7871e+5	5.2986e+3	4.7647e+4	8.7236e+3	5.1827e+3	2.2353e+3
	Std.	**22.2657**	6.0811e+3	5.8169e+2	1.1048e+4	771.5517	6.5217e+2	7.6850e+2
	Best	**80.9849**	1.6720e+5	4.37029e+3	2.6553e+4	7.2739e+3	3.7872e+3	1.0774e+3
	Rank	**1**	7	4	6	5	3	2
f_8 (D=1000)	Mean	4.7291e+4	1.2547e+7	3.8862e+5	1.8012e+6	1.5454e+6	2.6560e+4	**7.9139e+3**
	Std.	6.6212e+3	3.5539e+5	2.2329e+4	5.5949e+5	6.4511e+4	3.9984e+3	**1.4647e+3**
	Best	3.6574e+4	1.1381e+7	3.5017e+5	4.5453e+5	1.4470e+6	1.8651e+4	**4.4462e+3**
	Rank	3	7	4	5	6	2	**1**
f_9 (D=1000)	Mean	**9.7670e-9**	1.5616e+92	7.6578e-5	9.1116e+83	1.0103e+56	0.3859	2.2868e-6
	Std.	**3.3014e-8**	7.3622e+92	2.5647e-4	4.6589e+84	5.5321e+56	0.6475	1.7299e-6
	Best	**5.497e-16**	1.1184e+7	1.9892e-9	9.2943e+69	5.6083e+38	2.0490e-5	4.4192e-7
	Rank	**1**	5	2	7	6	4	3
Average Rank		2.555556	6.555556	3.777777778	5.777778	5.333333	2.222222	**1.666667**
Overall Rank		3	7	4	6	5	2	**1**

4.3.1.3 Convergence curves comparison of algorithms in unimodal benchmark functions

Fig. 2 is the fitness curves of the best fitness for unimodal benchmark functions $f1$-$f9$ (D =1000). As seen from Fig. 2, compared to the other six algorithms, the convergence of CSOA is faster, the precision of CSOA is better, except $f4$, $f7$, and $f9$. Although CSOA to $f9$ is worse than PSO in convergence and precision, CSOA has reached the theoretical best value, to $f7$ CSOA is only worse than PSO, to $f4$ CSOA is worse than PSO and SOA. Because the CSOA uses the multi-chain strategy to augment the individuals' diversity and local scout intensity, CSOA reveals better optimization property in unimodal functions.


f1 Sphere
f2 Schwefel2.22
f3 Rotated
f4 Schwefel2.21
f5 Rosenbrock
f6 Step
f7 Quartic
f8 SumSquares
PSO
SA+GA
GSA
SCA
MVO
SOA
CSOA
Fitness
Iteration
(a) enlarged image
(b) enlarged image
(c) enlarged image

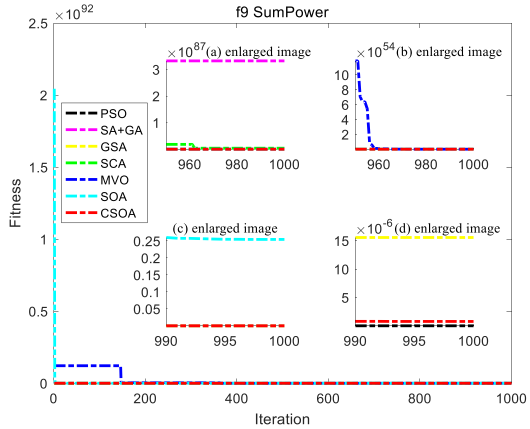


Fig. 2 Convergence curves for unimodal functions $f1$-$f9$(D =1000)

4.3.1.4 ANOVA tests comparison of algorithms in unimodal benchmark functions

Fig. 3 is the ANOVA of the global best values to unimodal functions $f1$-$f9$ (D =1000). As seen from Fig. 3, CSOA is the most robust, except $f3$, $f4$, and $f7$. The ANOVA test results of CSOA to $f7$ function is only worse than PSO algorithm, the result of CSOA to $f4$ is worse than PSO, GSA, and SOA, the result of CSOA to $f3$ function is only worse than GSA and SOA algorithm. The CSOA algorithm showed better robustness and improved SOA. Therefore, CSOA is an effective and feasible solution in the optimization of unimodal functions.

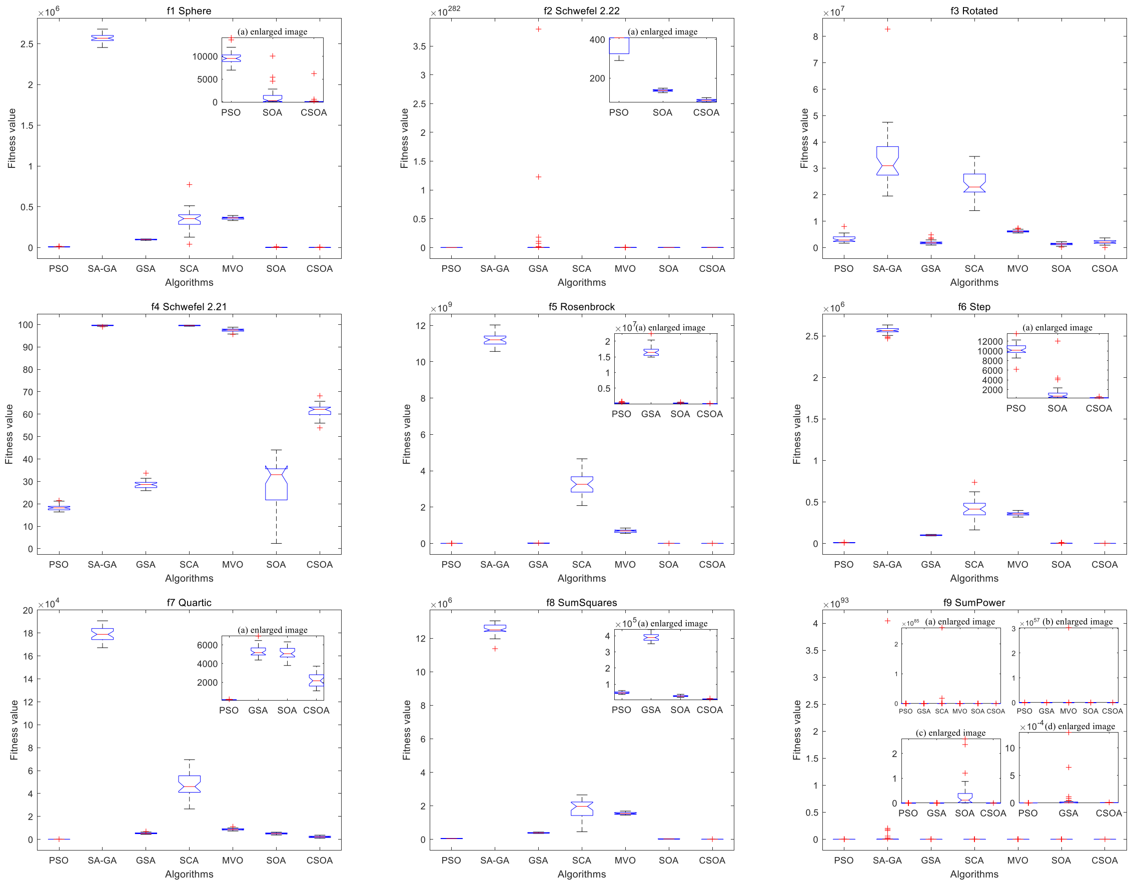


Fig. 3 ANOVA tests for unimodal functions $f1$-$f9$ (D =1000)

4.3.2 Algorithm performance comparison in multimodal benchmark functions

Same as above, it is common to base the capability of algorithms on mathematic functions that are known to be globally optimal. Following the same process, six multimodal reference functions were used as the comparative test platform [7,10,33-35].

4.3.2.1 Multimodal benchmark functions

Tab. 4 has shown the multimodal benchmark functions used in our experiment. The multimodal functions $f10$-$f15$ have many partly optimal values, which is useful for the algorithm to check, explore and avoid local optimal solutions. A thousand variables are considered for multimodal test functions for further improving their difficulties.

Tab. 4 Description of multimodal functions

Functions	Range	Minimum
$f_{10}(x) = -\sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	-418.9829* number of dimensions
$f_{11}(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
$f_{12}(x) = 20 + e - 20e^{-0.2}\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - e^{\frac{1}{n}}\sum_{i=1}^{n}\cos(2\pi x_i)$	[-32,32]	0
$f_{13}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
$f_{14}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ (-x_i - a)^m, x_i < -a \end{cases}$	[-50,50]	0
$f_{15}(x) = \frac{1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50,50]	0

4.3.2.2 Results comparison of algorithms in multimodal benchmark functions

Same as unimodal benchmark functions, the mean values, standard deviation, best fitness, and best fitness rank between the algorithms of 30 all alone runs, the multimodal functions $f10$-$f15$ are displayed in Tab. 5. The underline and boldface indicate that the optimal result is better.

For the multimodal benchmark functions, according to Tab. 5, except $f10$, $f11$, $f14$, and $f15$, to the optimal fitness, CSOA is better than others. The optimal fitness value results of CSOA to $f15$ function is only worse than PSO algorithm, the optimal fitness value result of CSOA to $f14$ function is worse than PSO and SOA algorithm, the optimal fitness value result of CSOA to $f11$ function is worse only than SCA algorithm, the result of CSOA to $f10$ is worse than SOA and MVO. Except $f10$, $f11$, $f12$, and $f14$, to the standard deviation results, CSOA are better than others, to $f14$ CSOA is worse than PSO and GSA, to $f11$ and $f12$ CSOA are worse than PSO, SA_GA, GSA, MVO, and SOA, the standard deviation results of CSOA to $f10$ function is worse than PSO, SA_GA, GSA, SCA, and MVO algorithm. Except $f10$ and $f14$, To the mean results, CSOA are better than others, to $f10$ CSOA is worse than MVO, and SOA, to $f14$ CSOA is only worse than PSO algorithm. According to the optimal fitness value mean rank and all rank results from Tab. 5, CSOA can find solutions, and the robustness is strong for multimodal benchmark functions.

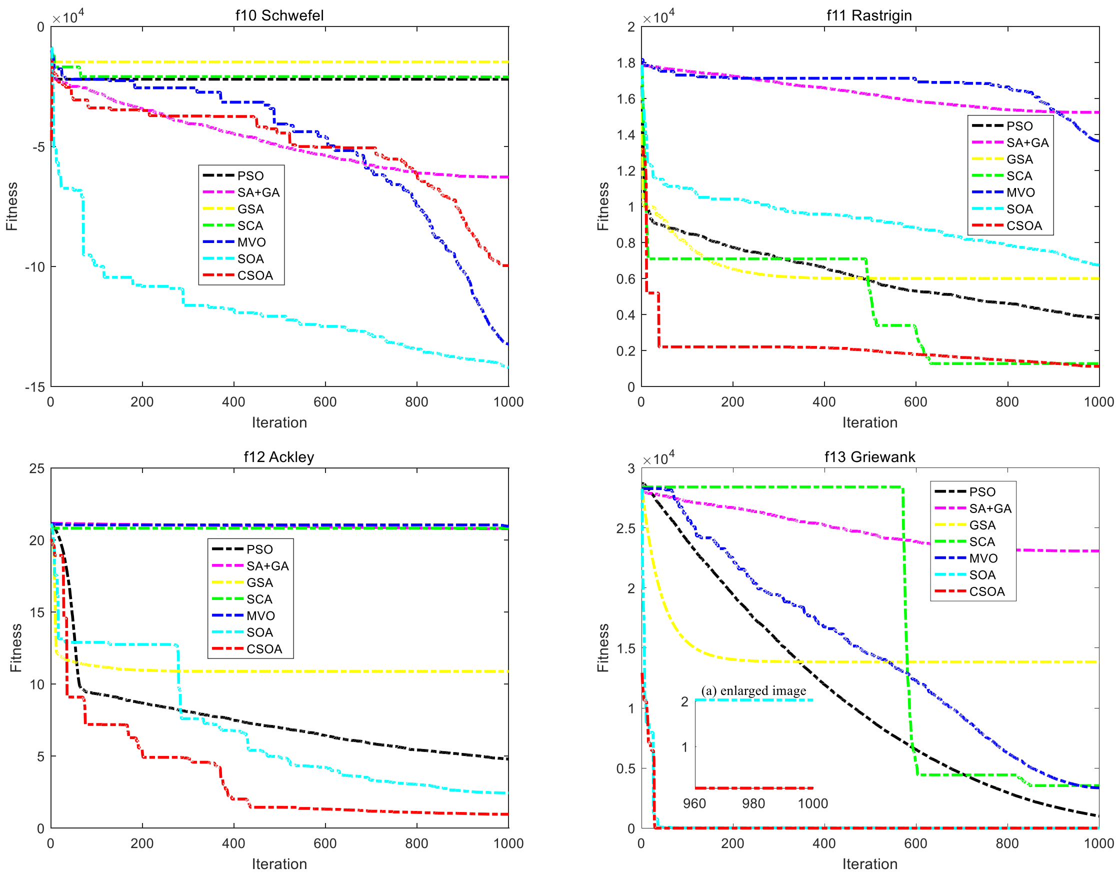
Tab. 5 Algorithms performance is compared of in multimodal functions

Functions	Result	Algorithm PSO	SA_GA	GSA	SCA	MVO	SOA	CSOA
f_{10} (D=1000)	Mean	-1.7007e+4	-5.8434e+4	-1.474e+4	-2.2849e+4	**<u>-1.339e+5</u>**	-1.181e+5	-7.995e+4
	Std.	2.5847e+3	3.4159e+3	2.3262e+3	**<u>1.3757e+3</u>**	5.9809e+3	3.0107e+4	1.5591e+4
	Best	-2.155e+4	-6.7056e+4	-1.844e+4	-2.665e+4	-1.467e+5	**<u>-2.157e+5</u>**	-1.178e+5
	Rank	6	4	7	5	2	**1**	3
f_{11} (D=1000)	Mean	2.8217e+3	1.5245e+4	5.7869e+3	1.8362e+3	1.3788e+4	6.3358e+3	**<u>1.7847e+3</u>**
	Std.	1.7027e+2	**<u>1.3944e+2</u>**	1.5766e+2	874.1338	300.8269	5.0727e+2	5.5996e+2
	Best	2.5372e+3	1.4955e+4	5.4142e+3	**<u>502.646</u>**	1.3250e+4	5.1713e+3	1.0788e+3
	Rank	3	7	5	**1**	6	4	2
f_{12}	Mean	4.4947	20.7977	10.3286	18.8639	20.9497	3.1246	**<u>1.5427</u>**
	Std.	0.1746	0.0267	0.1538	4.0427	**<u>0.0222</u>**	0.3837	0.9780

(*D* = 1000)	Best	4.2442	20.7492	9.9359	8.4914	20.8992	2.3556	**0.9571**
	Rank	3	6	5	4	7	2	**1**
f_{13} (*D* = 1000)	Mean	1.0210e+3	2.3100e+4	1.3996e+4	3.1433e+3	3.2496e+3	16.2019	**1.6504**
	Std.	32.8774	5.1266e+2	2.1571e+2	1.0529e+3	169.9815	27.1817	**5.2240**
	Best	9.5388e+2	2.1976e+4	1.3539e+4	1.1471e+3	2.8639e+3	0.19051	**0.0449**
	Rank	3	7	6	4	5	2	**1**
f_{14} (*D* = 1000)	Mean	**3.5824**	2.6235e+10	3.8689e+4	9.5072e+9	9.0076e+8	2.0865e+4	2.0766e+4
	Std.	**0.4677**	9.4178e+8	2.4728e+4	2.0642e+9	1.0468e+8	4.1982e+4	3.6384e+4
	Best	2.7478	2.4014e+10	5.6569e+3	5.1822e+9	7.1553e+8	**1.3229**	44.7877
	Rank	2	7	4	6	5	**1**	3
f_{15} (*D* = 1000)	Mean	1.4226e+4	4.9243e+10	6.1049e+6	1.6105e+10	2.2732e+9	9.6850e+4	**8.6561e+3**
	Std.	2.2790e+4	1.6004e+9	1.2137e+6	3.5103e+9	2.8365e+8	1.7378e+5	**1.2465e+4**
	Best	**4.0719e+2**	4.6279e+10	4.2910e+6	8.3903e+9	1.8130e+9	4.2437e+2	1.2094e+3
	Rank	**1**	7	4	6	5	3	2
Average Rank		3	6.333333	5.1666667	4.333333	5	2.166667	2
Overall Rank		3	7	6	4	5	2	1

4.3.2.3 Convergence curves comparison of algorithms in multimodal functions

Fig. 4 is the fitness curves of the global minimum values for multimodal benchmark functions $f10$-$f15$ (D =1000). As seen from Fig. 4, compared to the other six algorithms, the convergence of CSOA is faster, the precision of CSOA is better, except $f10$, $f14$ and $f15$. CSOA to $f15$ is only worse than SOA in convergence and precision aspect, to $f14$ CSOA function is worse than SOA and PSO, to $f10$ CSOA function is worse than MVO and SOA. CSOA reveals better optimization property in multimodal functions.



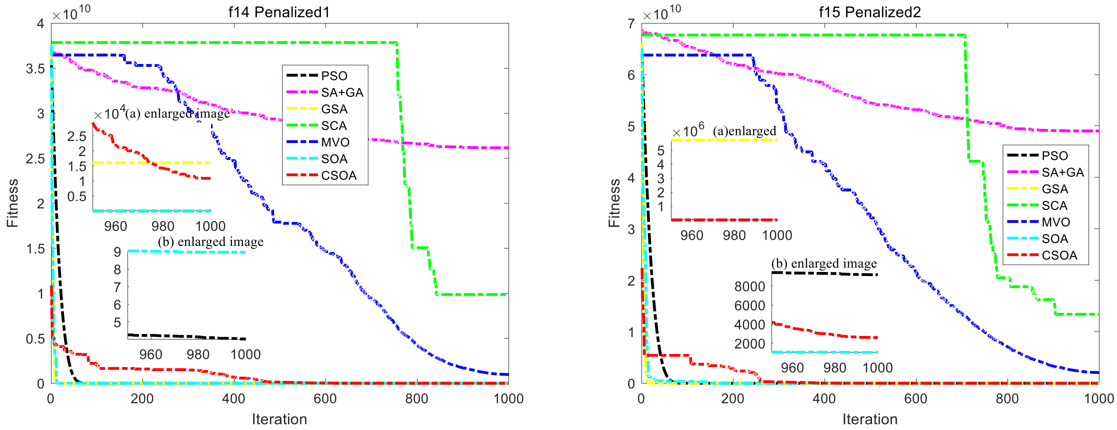


Fig. 4 Convergence curves for multimodal functions $f10$-$f15$ (D =1000)

4.3.2.4 ANOVA tests comparison of algorithms in multimodal benchmark functions

Fig. 5 is the ANOVA of the global best values to multimodal functions $f10$-$f15$ (D =1000). As seen from Fig. 5, CSOA is the most robust, except $f10$, $f12$ and $f14$. The ANOVA test results of CSOA to $f10$ function are only worse than the MVO algorithm, the results of $f12$ and $f14$ functions are worse than PSO and SOA algorithm. This CSOA algorithm showed better robustness and improves SOA. Therefore, CSOA is an effective and feasible solution in the optimization of multimodal functions.

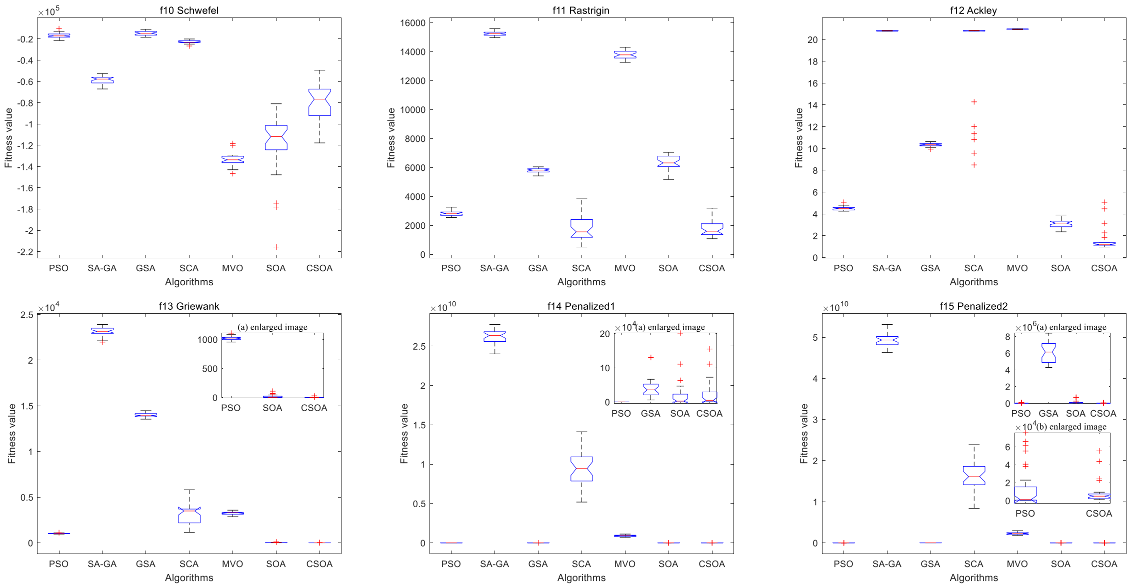


Fig. 5 ANOVA tests for multimodal functions $f10$-$f15$ (D =1000)

4.3.3 Complexity analysis

The calculational complexity of the basic SOA is O (N.D.M), N is the total individual count, D is the dimension count, M is the maximum count of algebras. The computational complexity of the first phase of the SOA stage is O (N.D.M). The complex coding strategy is introduced to calculate the O ((N.D.M) value. The introduced multi-chain strategy the calculational complexity value of O (N.D.M). So, the overall complexity of CSOA is O (N.D.M + N.D.M + N.D.M). Based on the principle of Big-O representation [36], if the count of algebras is high (M≫N, D), the calculational complexity is O ((N.D.M). Therefore, the overall calculational complexity of the CSOA is almost the same as the basic SOA.

4.3.4 Run time comparison of algorithms in benchmark functions

In the subsection, the running time of the algorithm for each function is recorded under the same conditions: population number 30, evolution algebra 1000, and 30 independent runs of the above fifteen benchmark functions $f1$-$f15$ (d=1000). Then, the running time of the fifteen functions was summed up to obtain the sum of the 30 independent running times of each algorithm for the fifteen functions listed in this paper. And rank the total time, as shown in Tab. 6. As can be seen from Tab. 6, the PSO algorithm has the most minor program running time, followed by the SCA algorithm, which has more minor program running time, and the CSOA algorithm ranks sixth, which has relatively more program running time. At the bottom of the list is the SA_GA algorithm, which takes the most running time.

Tab. 6 Run time comparison of 30 independent runs for benchmark functions $f10$-$f15$ (D =1000)

Functions (D= 1000)	Run time of algorithms						
	PSO	SA_GA	GSA	SCA	MVO	SOA	CSOA
f_1	100.719730	880.039134	1238.641454	123.011947	291.672145	151.787862	1240.526227
f_2	68.229668	848.231807	1319.453189	158.694010	120.413171	201.005115	1120.092672
f_3	2083.966419	59198.227940	3001.030713	1813.227548	1978.830114	6464.103793	19353.182034
f_4	55.090061	763.100004	1182.766535	148.610507	282.693611	127.111629	991.587944
f_5	59.726643	907.492580	1190.101594	133.724941	347.492206	171.346819	1036.987842
f_6	60.136433	768.869660	1321.559282	127.490008	283.733081	137.581005	1054.889547
f_7	171.086743	2893.833975	1345.984756	203.877433	366.515217	378.499704	1809.255687
f_8	56.468720	824.393464	1361.309649	131.849271	296.880364	141.850056	1226.496492
f_9	119.475367	2814.347692	1338.479782	202.890012	270.227397	373.595086	1727.571955
f_{10}	96.665578	1331.028369	1207.890758	162.894968	163.620252	300.359964	1248.552418
f_{11}	67.987998	1166.055564	1200.353798	147.172828	321.039558	243.508636	1135.691572
f_{12}	85.468484	1298.972295	1504.250466	168.363508	421.893238	329.211669	1409.548409
f_{13}	105.918299	1489.631271	1217.56270	147.627982	330.277401	224.838827	1231.581318
f_{14}	227.567551	5817.019363	1364.117264	381.158993	485.116503	782.243419	2873.449473
f_{15}	252.124024	5953.028083	1367.575562	320.156020	494.213399	1149.480076	2800.126364
The total time	**3610.632**	86954.27	21161.08	4370.74998	6454.6177	11176.52	40259.54
Overall Rank	**1**	7	5	2	3	4	6

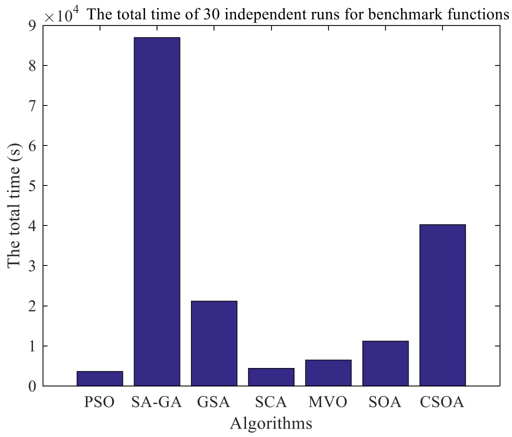


Fig. 6 The total time of 30 independent runs of 7 algorithms on 15 benchmark functions

To learn more about the program running time of the seven algorithms in the fifteen functions, a bar chart Fig. 6 was made for the total time of each algorithm after 30 independent runs. From Fig. 6, the program running time of PSO is the least, while that of SA_GA algorithm is the most, the program running time of CSOA is less than half of that of SA_GA algorithm, which is relatively large.

4.3.5 Performance profiles of algorithms in benchmark functions

The average fitness was selected as the capability index. The algorithmic capability is expressed in performance profiles, which are calculated by the formulas (18)(19).

$$r_{f,g} = \mu_{f,g} / \min\{\mu_{f,g} : g \in G\} \quad (18)$$

$$\rho_g(\tau) = \text{size}\{ f \in F : r_{f,g} \le \tau \} / n_f \quad (19)$$

Where, g represents an algorithm, G is the algorithms set, f means a function, F is the function set, n_g is the number of algorithms in the experiment, n_f is the number of functions in the experiment, $\mu_{f,g}$ is the average fitness after algorithm g solving function f, $r_{f,g}$ is the capability ratio, ρ_g is the algorithmic capability, τ is a factor of the best probability [37].

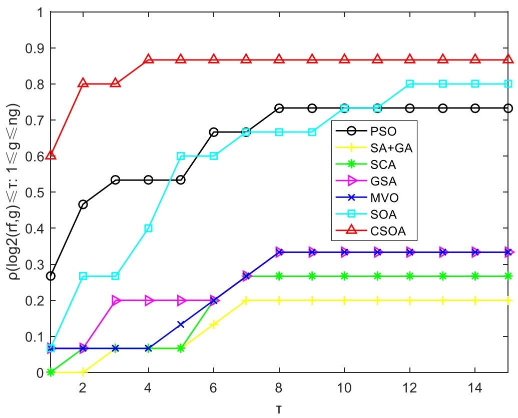


Fig. 7 Performance profile of 7 algorithms on 15 benchmark functions

Fig. 7 shows the capability ratios of the mean fitness for the seven algorithms on the benchmark functions $f1$- $f15$ (D =1000). The results are displayed by a log scale 2. As shown in Fig. 7, CSOA has the highest probability. When τ = 1 CSOA is about 0.6, which is better than others. When τ = 4 CSOA is about 0.87, PSO is 0.53, SOA is 0.40, GSA is 0.067, MVO is 0.067, SCA is 0.067, SA_GA is 0.067. When τ = 12 CSOA is 0.87, PSO is 0.73, SOA is 0.80, GSA is 0.33, MVO is 0.33, SCA is 0.27, SA_GA is 0.2. The capability curve of CSOA lies above others, and CSOA can achieve about 87% when τ ≥ 4. CSOA performs obviously better than other algorithms.

4.4 Algorithm performance comparison in PID controller parameter optimization problems

This subsection, using four test control system models optimizing PID parameters to test the capability of the CSOA algorithm. For g1~g3, population number of all algorithms is 20, the max number of algebras is 20, g1~g2 step response time set 10s, g3 step response time set 30s. For g4 population number of all algorithms is 50, the max number of algebras is 50, step response time is set at 50s.

4.4.1 Control system models

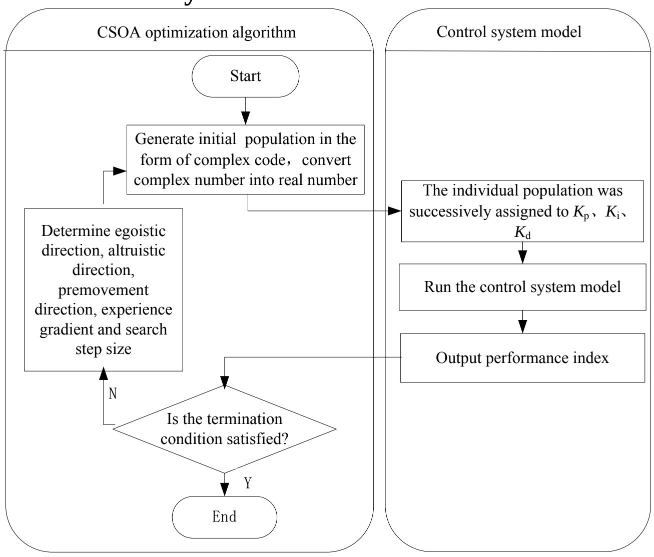


Fig. 8 A process diagram for optimizing test control system PID parameters by CSOA

Equations 20-23 have shown test control system models optimizing PID parameters used in our experiment. Fig. 8 have shown the process diagram for optimizing test control system PID parameters by CSOA. Fig. 9 have shown the optimization PID parameters model structure of the test control system.

$$g_1(s) = \frac{2.6}{(2.7s+1)(0.3s+1)} \tag{20}$$

$$g_2(s) = \frac{5}{(2.7s+1)} e^{-0.5s} \tag{21}$$

$$g_3(s) = \frac{3}{(2s+1)} e^{-3s} \tag{22}$$

$$g_4(s) = \frac{1}{(s+1)^8} \tag{23}$$

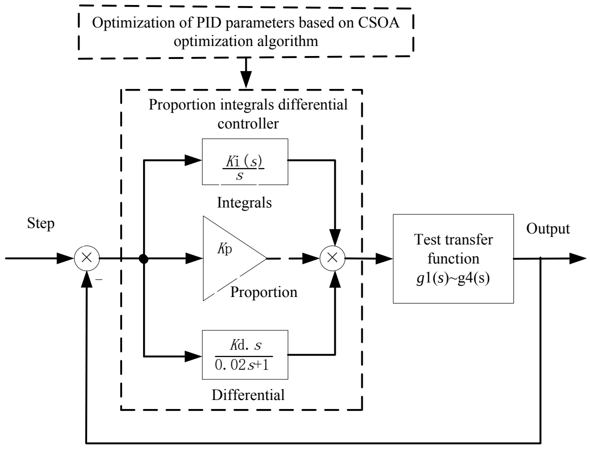


Fig. 9 Optimization PID parameters model structure of test control system

4.4.2 Results comparison of algorithms in PID controller parameter optimization

For testing the capability of the CSOA algorithm, CSOA has compared with PSO, SA-GA, GSA, SCA, MVO, and SOA in PID controller parameters optimization. The mean values, standard deviation values, best fitness values, and best fitness values rank between the algorithms of 30 all alone runs, for g1~g4 are displayed in Tab. 7. The underline and boldface indicate that the optimal result is better.

For the PID controller parameter optimization problems, according to Tab. 7, except g3, and g4, to the best fitness, CSOA is better than others. The optimal fitness value results of the CSOA to g3 model are only worse than the SA_GA algorithm; the optimal fitness value result of the CSOA to g4 model is only worse than the PSO algorithm. Except for g2, and g3, To the standard deviation results, CSOA is better than others, CSOA is only worse than SA_GA. To the mean, CSOA is better than others. According to the optimal fitness value mean rank and all rank results from Tab. 7, CSOA can find solutions and has very strong robustness for PID controller parameter optimization problems.

Tab. 7 Performance comparison of algorithms in PID parameter optimization of 30 independent runs

Test	Result	Algorithm						
		PSO	SA_GA	GSA	SCA	MVO	SOA	CSOA
g_1	Mean	0.2267	0.3169	0.4571	0.0918	0.2501	0.1917	**<u>0.0539</u>**
	Std.	0.0877	0.0649	0.1569	0.0263	0.0532	0.11226	**<u>0.0127</u>**
	Best	0.0485	0.1002	0.2732	0.0483	0.0513	0.05774	**<u>0.0479</u>**

	Rank	3	6	7	2	4	5	**1**
g2	Mean	58.4757	62.4599	60.7787	24.8454	59.5805	42.1538	**0.8233**
	Std.	7.75976	**0.1216**	5.3034	21.5239	7.6556	27.9025	0.6631
	Best	36.0409	62.0356	42.7711	0.4898	32.6095	0.39301	**0.3299**
	Rank	5	7	6	3	4	2	**1**
g3	Mean	1.8481e+2	2.7179e+2	2.7665e+2	29.0458	1.0848e+2	2.6269e+2	**13.0293**
	Std.	59.6434	**0.62334**	10.3088	11.9839	56.6750	44.8106	1.7189
	Best	32.5445	**2.71191**	2.7139e+2	14.5588	20.0492	26.5763	10.1561
	Rank	6	**1**	7	3	4	5	2
g4	Mean	1.7713e+2	55.3556	2.3413e+2	85.196656	35.721213	46.10528	**34.63334**
	Std.	4.2182e+2	36.00807	2.1754e+2	1.0050e+2	1.411226	26.992197	**0.00686**
	Best	**34.625063**	34.6294	58.321733	34.867448	34.643162	34.745734	34.625096
	Rank	**1**	3	7	6	4	5	2
Average Rank		3.75	4.25	6.75	3.5	4	4.25	1.5
Overall Rank		3	5	7	2	4	5	**1**

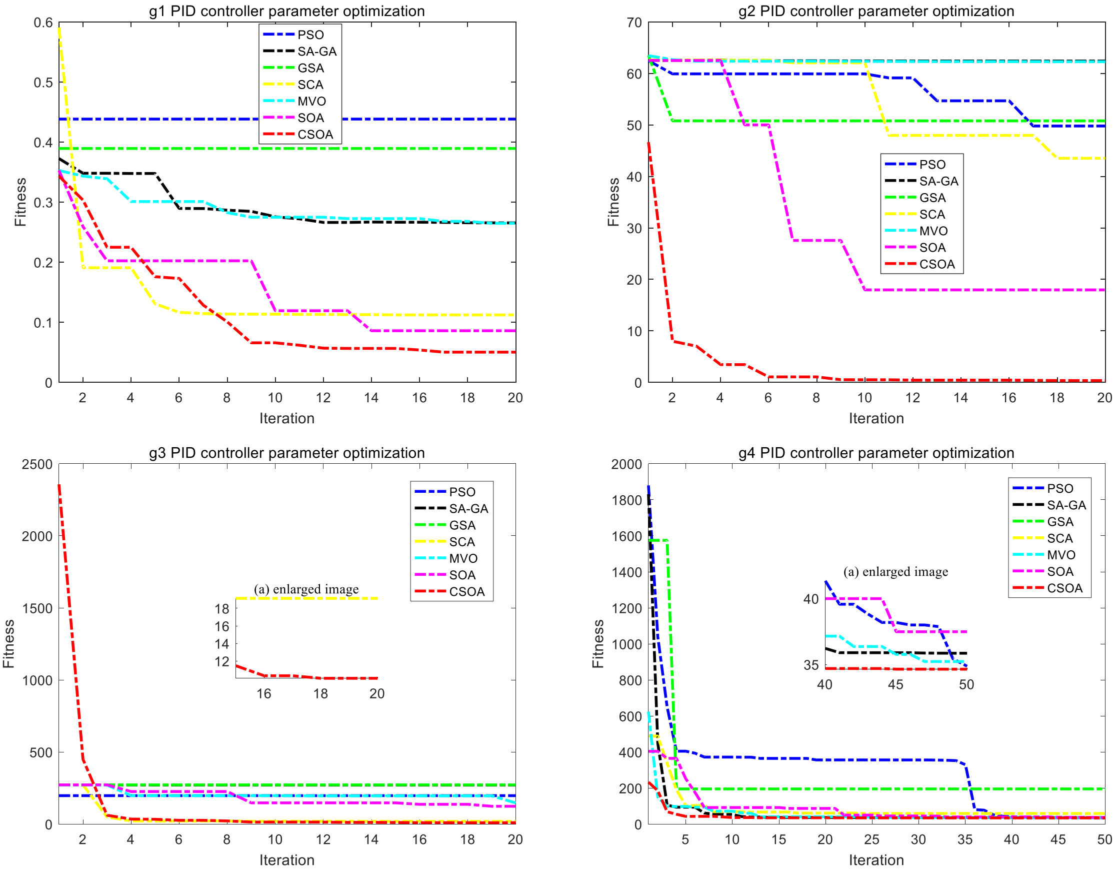
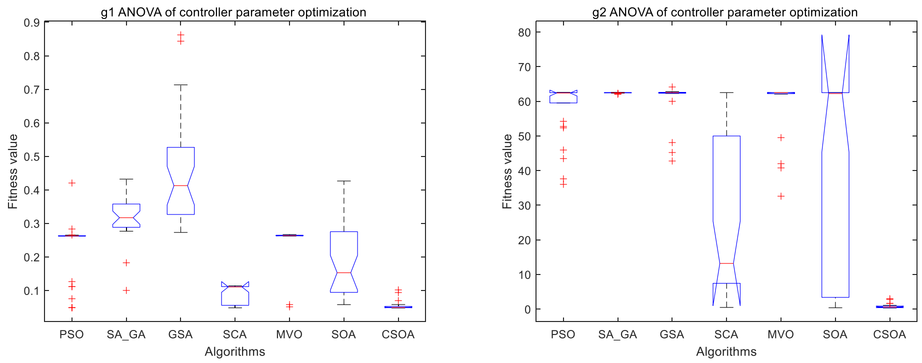


Fig. 10 Convergence curves for PID controller parameter optimization g1 – g4



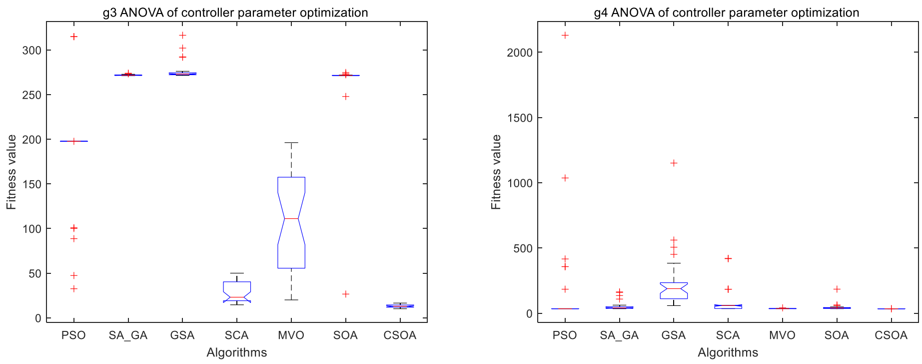


Fig. 11 The ANOVA tests for PID controller parameter optimization g1 - g4

4.4.3 Convergence curves comparison of algorithms in PID controller parameter optimization

Fig. 10 have shown the fitness curves PID controller parameter optimization for g1~g4. As shown in Fig. 10, compared to the other six algorithms, the convergence of the CSOA is fast, the precision of the CSOA is best. CSOA can find the optimal value.

4.4.4 ANOVA tests comparison of algorithms in PID controller parameter optimization

Fig. 11 is the ANOVA of the global best values PID controller parameter optimization for g1~g4. As seen from Fig. 11, CSOA is the most robust than other algorithms.

4.4.5 The unit step functions PID controller parameter optimization

Fig. 12 have shown the unit step functions PID controller parameter optimization for g1~g4. As seen from Fig. 12, by CSOA algorithm to optimization unit step models PID controller parameter for g1–g4, unit step functions tend to stabilize very quickly and accurately.

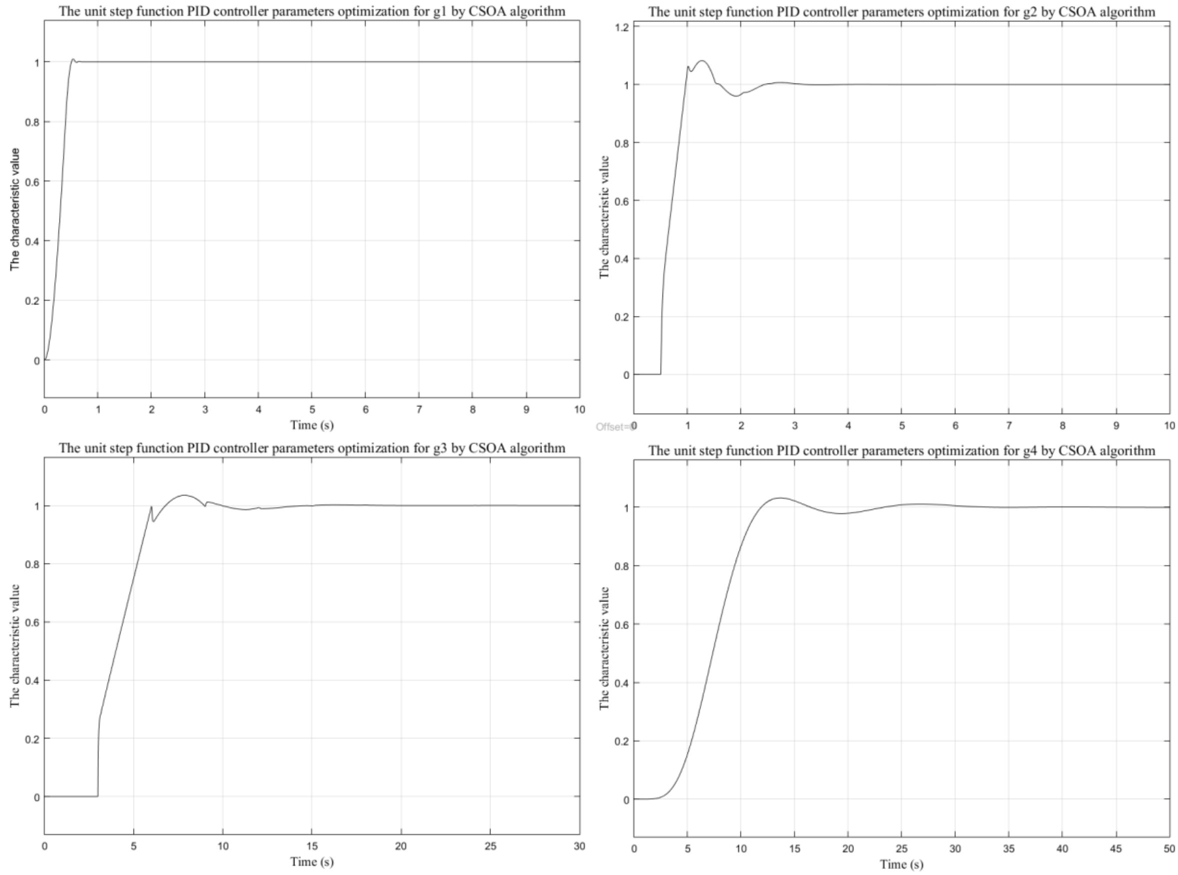


Fig. 12 The unit step functions PID controller parameter optimization g1~g4

Therefore, CSOA is an effective and feasible solution in control system models optimizing PID parameters by algorithms.

4.5 Algorithm performance comparison in constrained engineering optimization problems

We are using six engineering problems to test the capability of the CSOA algorithm further. That engineering problems are very popular in the literature. The penalty function is used to calculate the constrained problem. The parameters set for all heuristic algorithms still adopts the parameter setting of section 4.2.

4.5.1 Welded beam design problem

This is a lest fabrication cost problem, which has four parameters and seven constraints. The parameters of the structural system are shown in Fig. 13[9]. The formulations of this problem are available in Appendix.

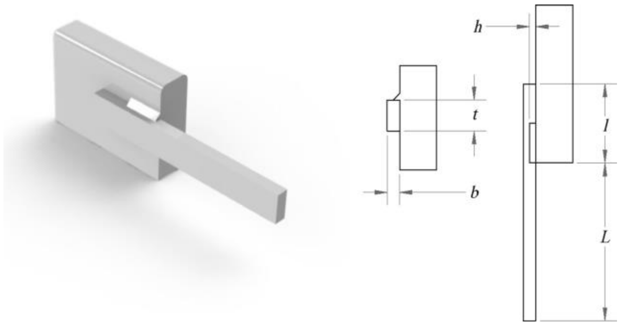


Fig. 13 Design parameters of the welded beam

Tab. 8 Comparison results of the welded beam design problem

Algorithm	Optimal values for variables				Optimal cost	Rank
	h	*l*	*t*	*b*		
GSA [6]	0.182129	3.856979	10.0000	0.202376	1.87995	9
MFO [7]	0.2057	3.4703	9.0364	0.2057	1.72452	5
MVO [9]	0.205463	3.473193	9.044502	0.205695	1.72645	6
CPSO [38]	0.202369	3.544214	9.048210	0.205723	1.72802	7
HS [39]	0.2442	6.2231	8.2915	0.2443	2.3807	12
PSO	0.20437461682	3.27746206207	9.03907307954	0.20573458497	1.69700648019	2
SA-GA	0.26572876298	2.77789863579	7.63164040030	0.28853829376	1.99412873170	10
GSA	0.12743403146	5.89076184871	8.05262845397	0.25908004232	2.10212926568	11
SCA	0.20112344041	3.23948182622	9.40574225336	0.20795790595	1.76704865429	8
MVO	0.20397627841	3.28970350716	9.03536739179	0.20582407425	1.69811381975	4
SOA	0.19348578918	3.489546622637	9.027709656861	0.20615302629	1.69714450048	3
CSOA	0.20568280035	3.25692824444	9.03941142183	0.20578118608	**1.69655946036**	**1**

Some of the works comes from the literature: GSA [6], MFO [7], MVO [9], CPSO [38], and HS [39]. This paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best-obtained values in Tab. 8.

From Tab. 8, compared with mathematical methods, this algorithm has the advantage of avoiding local optimality. For the problem, the CSOA algorithm is better than GSA, MFO, MVO, GA, CPSO, and HS algorithms in other kinds of literature. The CSOA is also better than PSO, SA_GA, GSA, SCA, MVO, and SOA. Therefore, CSOA is an effective and feasible solution to the problem.

4.5.2 Pressure vessel design problem

This is a lest fabrication cost problem of four parameters and four constraints. The parameters of the structural system are shown in Fig. 14 [9]. The formulations of this problem are available in Appendix.

Some of the works comes from the literature: MFO [7], ES [40], DE [41], ACO [42], and GA [43]. In this paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best-obtained values in Tab. 9.

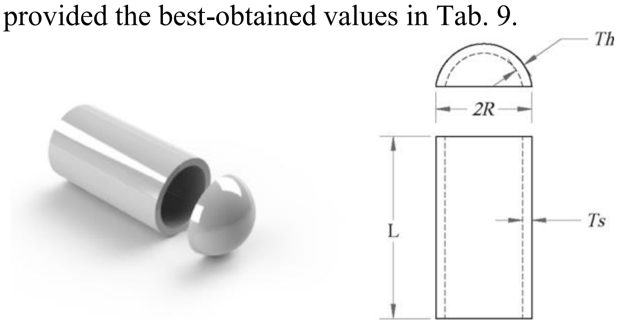


Fig. 14 Pressure vessel design problem

From Tab. 9, compared with mathematical methods, this algorithm has the advantage of avoiding local optimality. For the problem, the CSOA algorithm is better than MFO, ES, DE, ACO, and GA algorithms in other kinds of literature. The CSOA is also better than PSO, SA_GA, GSA, SCA, MVO, and SOA. Therefore, CSOA is an effective and feasible solution to the problem.

Tab. 9 Comparison results for pressure vessel design problem

Algorithm	Optimal values for variables				Optimal cost	Rank
	T_s	T_h	*R*	*L*		
MFO [9]	0.8125	0.4375	42.098445	176.636596	6059.7143	8
ES [40]	0.8125	0.4375	42.098087	176.640518	6059.7456	10
DE [41]	0.8125	0.4375	42.098411	176.637690	6059.7340	9
ACO [42]	0.8125	0.4375	42.103624	176.572656	6059.0888	7
GA [43]	0.8125	0.4375	42.097398	176.654050	6059.9463	11
PSO	0.93627266112	0.41391783346	47.19019859907	123.06285131625	6317.0167340514	12
SA-GA	0.83804097369	0.41223740796	45.10610463950	142.64078515697	5931.2868373440	5

GSA	0.89533101776	0.43654377356	47.89640596198	115.96279725902	6057.9309555313	6
SCA	0.71165237901	0.39215740603	40.39056304889	200.00000000000	5903.0036698882	4
MVO	0.75462696023	0.37830685291	40.94839768196	191.64503059607	5764.4347452930	3
SOA	0.76961590364	41.5284631287	0.388196715944	183.84147207932	5735.1355906012	2
CSOA	0.74366609133	40.3200509108	0.366463323335	200.00000000000	**5735.0844852589**	**1**

4.5.3 Cantilever beam design problem

This is a problem that is determined by five parameters and applied only to the scope of variables of constraints. The parameters of the structural system are shown in Fig. 15 [7]. The formulations of this problem are available in Appendix.

Some of the works comes from the literature: MFO [7] CS [44], GCA [45], MMA [45], and SOS [46]. In this paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best-obtained values in Tab. 10.

From Tab. 10, the CSOA algorithm is better than MFO, CS, GCA, MMA, and SOS algorithms in other kinds of literature. The CSOA is also better than PSO, SA_GA, GSA, SCA, MVO, and SOA. Therefore, CSOA is an effective and feasible solution to the problem.

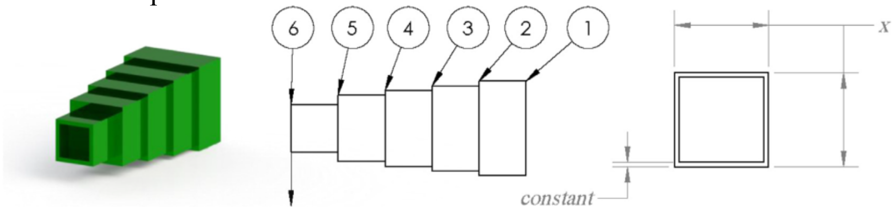


Fig. 15 Cantilever beam design problem

Tab. 10 Comparison results for cantilever beam design problem

Algorithm	Optimal values for variables					Optimum weight	Rank
	x_1	x_2	x_3	x_4	x_5		
MFO [7]	5.9848717732	5.3167269243	4.4973325858	3.5136164677	2.1616202934	1.339988086	6
CS [44]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999	7
GCA [45]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	8
MMA[45]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	8
SOS [46]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996	3
PSO	6.007219438	5.311747232	4.505611438	3.4904346887	2.158626706	1.339963522	4
SA-GA	6.251285023	5.460509756	4.149903306	3.8032391760	1.974102742	1.350285757	11
GSA	6.020285873	5.305304583	4.512114944	3.4939372220	2.142187864	1.339969652	5
SCA	5.801308754	5.589807963	4.497563735	3.4994713866	2.262668613	1.351011196	12
MVO	6.017944991	5.336576175	4.493102726	3.4797461041	2.146292918	1.340024388	10
SOA	6.014092415	5.315583298	4.484154000	3.5033360363	2.156331174	1.339957455	2
CSOA	6.013067642	5.292274798	4.491025571	3.5095364396	2.167826747	**1.339954592**	**1**

4.5.4 Gear train design problem

This is a minimize the gear ratio problem, which has four variables and the scope of variables of constraints. Fig. 16 is the schematic diagram [47]. The formulations of this problem are available in Appendix.

Some of the works comes from the literature: MFO [7], MVO [9], CS [44], ABC [48], MBA [48]. In this paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best obtained-values in Tab. 11.

From Tab. 11, CSOA algorithm are better than MFO, MVO, CS, ABC, and MBA algorithms in other kinds of literature. Except for SA_GA, GSA, and PSO, the CSOA also better than SCA, MVO, and SOA. The optimal fitness value of CSOA has reached the theoretical best value, although the optimal fitness value of CSOA is worse than SA_GA, GSA, and PSO. CSOA finds a new value. Therefore, CSOA is an effective and feasible solution to the problem.

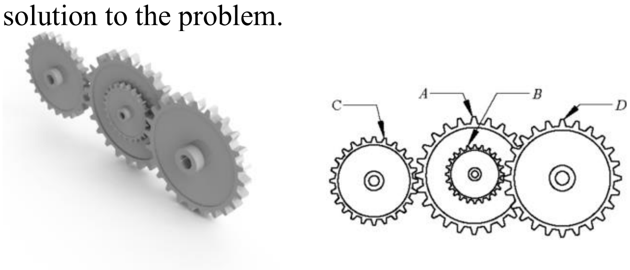


Fig. 16 Gear train design problem

Tab. 11 Comparison results of the gear train design problem

Algorithm	Optimal values for variables				Optimal gear ratio	Rank
	n_A	n_B	n_C	n_D		
MFO [7]	43	19	16	49	2.7009e-012	7

MVO [9]	43	16	19	49	2.7009e-012	7
CS [44]	43	16	19	49	2.7009e-012	7
ABC [48]	49	16	19	43	2.7009e-012	7
MBA [48]	43	16	19	49	2.7009e-012	7
PSO	41.2676387267	12.0000000000	12.0000000000	24.1851491677	5.321647791e-20	3
SA-GA	32.3132176916	21.0818982120	12.1649288759	55.0091556193	**0**	**1**
GSA	54.7718113206	33.5951575204	12.0000000000	51.0148628266	1.358936169e-30	2
SCA	52.6322252242	15.4114043064	23.1179418870	46.9168162381	5.431797718e-12	12
MVO	60.0000000000	12.0000000000	41.8647883833	58.0329758032	2.334953506e-16	5
SOA	60.0000000000	12.0000000000	43.2835302093	60.0000000000	2.567448245e-16	6
CSOA	43.3821083557	31.2957045927	12.0000000000	60.0000000000	7.763414089e-17	4

4.5.5 Three-bar truss design problem

This is a minimize weight problem under stress, which has two variables and applying only to the scope of variables of constraints. The schematic diagram of components [47] is shown in Fig. 17[9]. The formulations of this problem are available in Appendix.

Some of the works comes from the literatures: MFO [7], MVO [9], CS [44], MBA [48], DEDS [49]. In this paper, the problem is resolved by the CSOA and compared to PSO, SA_GA, GSA, SCA, MVO, and SOA. In this paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best-obtained values in Tab. 12.

From Tab. 12, except MVO, and PSO, the CSOA algorithm is better than others. The optimal fitness value of CSOA has reached the theoretical best value, although the optimal fitness value of CSOA is worse than MVO and PSO. Therefore, CSOA is an effective and feasible solution to the problem.

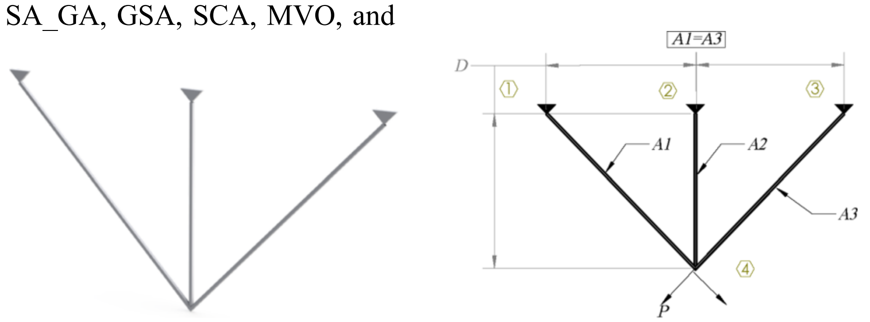


Fig. 17 Three-bar truss design problem

Tab. 12 Comparison results of the three-bar truss design problem

Algorithm	Optimal values for variables		Optimum weight	Rank
	x_1	x_2		
MFO [7]	0.788244770931922	0.409466905784741	263.895979682	10
MVO [9]	0.78860276	0.40845307	263.8958499	8
CS [44]	0.78867	0.40902	263.9716	11
MBA [48]	0.7885650	0.4085597	263.8958522	9
DEDS [49]	0.78867513	0.40824828	263.8958434	7
PSO	0.788425434690935	0.408085596065985	263.8523465301364	2
SA-GA	0.787321758816231	0.411216143996852	263.8532291023197	5
GSA	0.761893501005708	0.493138841375638	264.8099085788021	12
SCA	0.789922169365255	0.403817724788810	263.8541885386347	6
MVO	0.788407496115311	0.408135122885127	**263.8523464859033**	**1**
SOA	0.788530250484097	0.407914579681955	263.8523714388302	4
CSOA	0.788444195859439	0.408029807190657	263.8523473086418	3

4.5.6 I-beam design problem

This is a minimize vertical deflection problem that has four variables and a constraint. Fig. 18 is the design diagram [7]. The formulations of this problem are available in Appendix.

Some of the works comes from the literature: MFO [7], CS [44], SOS [46], IARSM [50], and ARSM [50]. In this paper, about the problem the CSOA compared to PSO, SA_GA, GSA, SCA, MVO, and SOA, and provided the best-obtained values in Tab. 13.

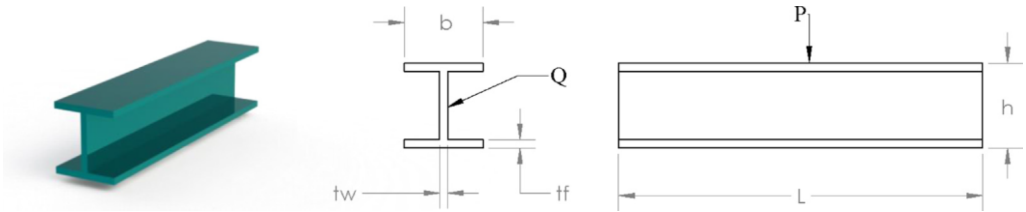


Fig. 18 I-beam design problem

Tab. 13 Comparison results for I-beam design problem

Algorithm	Optimal values for variables				Optimum vertical deflection	Rank
	b	h	t_w	t_f		
MFO [7]	50	80	1.7647	5.0000	**0.0066259**	**1**
CS [44]	50	80	0.9 5	2.32167	0.0130747	9
SOS [46]	50	80	0.9	2.32179	0.0130741	8
IARSM[50]	48.42	79.99	0.90	2.40	0.131	11
ARSM [50]	37.05	80	1.71	2.31	0.0157	10
PSO	29.2349505988	77.7790428198	5.0000000000	3.5987373218	0.0114625520	12
SA-GA	34.9999839459	79.9999646294	4.9999802368	4.9999823841	0.0078637302	4
GSA	35.0000000001	80.0000000000	5.0000000000	5.0000000000	0.0078636959	2
SCA	34.9878089422	80.0000000000	5.0000000000	5.0000000000	0.0078658199	7
MVO	34.9998614894	80.0000000000	4.9997841775	5.0000000000	0.0078637964	6
SOA	34.9999002914	80.0000000000	5.0000000000	5.0000000000	0.0078636963	3
CSOA	34.9997858604	80.0000000000	5.00000000000	5.0000000000	0.0078637332	5

In Tab. 13, except MFO, GSA, SOA, and SA-GA, the CSOA algorithm are better than others. The fitness of MFO is best. Although the most minor vertical deviation of the CSOA algorithm is not as good as that of GSA, SOA, and SA-GA algorithms, it is very close to other relative optimal values. Therefore, CSOA is an effective and feasible solution to the I-beam design optimization problem.

In brief, the CSOA algorithm fulfills better than other algorithms in most actual studies. CSOA is an effective and feasible solution to the practical optimization problems.

5 Conclusion

A CSOA algorithm is presented, with a complex value encoding method and a multi-link strategy. According to four phases test to CSOA from different perspectives: unimodal benchmark functions, multimodal benchmark functions, PID control parameters, and constrained engineering. Besides, CSOA was compared to PSO, SA-GA, GSA, SCA, MVO, and SOA.

In the first phase, CSOA was test in nine benchmark functions. The results are that the CSOA algorithm is very effective and feasible in unimodal functions. In the phase, we consider the ranking values of 30 all alone running between CSOA mean values, standard deviation values, best fitness values, and best fitness values rank, convergence curves, and variance tests for global minimum values.

In the scend test phase, six multimodal benchmark function optimization problems were used to test CSOA further. The CSOA algorithm is also very effective and feasible in multimodal functions. The second test phase also considered the ranking values of 30 all alone running between CSOA mean values, standard deviation values, best fitness values, and best fitness values rank, convergence curves, and variance tests for global minimum values.

From the nine unimodal benchmark functions and six multimodal benchmark functions optimization problems, the overall calculational complexity of the CSOA is almost the same as the basic SOA; the run time comparison of seven algorithms in benchmark functions, CSOA algorithm has relatively more program running time, it is not optimal about running time; to the capability of seven algorithms, CSOA is the highest competitive algorithm.

In the third test phase, four PID control parameter optimization was used to test CSOA in practice further. The problems were parameter optimization model of second-order PID controller without time delay, parameter optimization model of PID controller with first order micro-delay, parameter optimization model of first-order PID controller with significant time delay, and parameter optimization model of high order PID controller without time delay problems. The third test phase also considered CSOA mean values, standard deviation values, best fitness values, and best fitness values rank of 30 all alone runs, the convergence curves, and ANOVA. From the results of PID parameters optimization problems, compared to the other six algorithms, CSOA is effective and feasible in practical problem.

Eventually, in the last test phase, six engineering problems further tested CSOA. The CSOA was compared to various algorithms. The results displayed CSOA is the highest competitive algorithm in the practical optimization problems .

According to comparative analysis of experiments, the conclusion is as follow:

- Using complex-valued encoding to each seeker increases the scout region and avoids convergence to local optimality.
- Using the randomly generated multi-chain strategy increases the search space.
- Multi-chain tends to generate in any seeker randomly, so increasing the diversity of seeker.
- CSOA optimization benchmark function has higher optimization capability.
- CSOA optimization benchmark function has almost the same calculational complexity as the SOA.
- The running time of the CSOA optimization benchmark function is relatively high. Among the seven algorithms compared, the running time is only better than that of the SA algorithm.
- The CSOA can solve real challenging problems, such as: PID control parameters optimization problems, and classical constrained engineering optimization problems.
- Further tuning and adjustments can be incorporated into future studies.

Acknowledgments This work was supported by National Science Foundation of China (Grant No. 51766005) and Science and technology project of Yunnan tobacco company of China (Grant No. 2019530000241019).

Appendix

I – Welded beam design problem

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

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

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

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

$g_3(\vec{x})=x_1-x_4\le 0$,

$g_4(\vec{x})=1.10471x_1^2+0.04811x_3x_4(14+x_2)-5\le 0$,

$g_5(\vec{x})=0.125-x_1\le 0$,

$g_6(\vec{x})=\delta(\vec{x})-\delta_{\max}\le 0$,

$g_7(\vec{x})=P-P_c(\vec{x})\le 0$,

Variable range $0.1\le x_1\le 2$, $0.1\le x_2\le 10$, $0.1\le x_3\le 10$, $0.1\le x_4\le 2$,

Where $\tau(\vec{x})=\sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2}$,

$\tau'=\frac{P}{\sqrt{2}x_1x_2}$, $\tau''=\frac{MP}{J}$, $M=P(L+\frac{x_2}{2})$,

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

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

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

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

P=6000 lb, L=14 in, $E=30\times10^6$ psi, $G=12\times10^6$ psi,
$\tau_{\max}$=136000 psi, $\sigma_{\max}$=30000 psi, $\delta_{\max}$=0.25 in.

II–Pressure vessel design problem

Consider $\vec{x}=[x_1,x_2,x_3]=[T_s,T_h,R,L]$,

Minimize

$f(\vec{x})=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1661x_1^2x_4+19.84x_1^2x_3$,

Subject to $g_1(\vec{x})=-x_1+0.0193x_3\le 0$,

$g_2(\vec{x})=-x_2+0.00954x_3\le 0$,

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

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

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

III–Cantilever design problem

Consider $\vec{x}=[x_1,x_2,x_3,x_4,x_5]$,

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

Subject to $g(\vec{x})=\frac{61}{x_1^3}+\frac{37}{x_2^3}+\frac{19}{x_3^3}+\frac{7}{x_4^3}+\frac{1}{x_5^3}-1\le 0$, Variable range $0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100$.

IV –Gear train design problem

Consider $\vec{x} = [x_1, x_2, x_3, x_4] = [n_A, n_B, n_C, n_D]$,

Minimize $f(\vec{x}) = (\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4})^2$,

Variable range $12 \le x_1,\ x_2,\ x_3,\ x_4 \le 60$.

V–Three-bar truss design problem

Consider $\vec{x} = [x_1, x_2] = [A_1, A_2]$,

Minimize $f(\vec{x}) = (2\sqrt{2}x_1 + x_2) * l$,

Subject to $g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P\text{-}\sigma \le 0$,

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P\text{-}\sigma \le 0$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1} P\text{-}\sigma \le 0$$

Variable range $0 \le x_1,\ x_2 \le 1$, $l = 100\text{cm}$, $P = 2\text{KN}/\text{cm}^2$, $\sigma = 2\text{KN}/\text{cm}^2$.

VI–I-beam design problem

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

Minimize
$$f(\vec{x}) = \frac{5000}{\frac{x_3(x_2 - 2x_4)^3}{12} + \frac{x_1 x_4^3}{6} + 2x_1 x_4 \left(\frac{x_2 - x_4}{2}\right)^2},$$

Subject to $g(\vec{x}) = 2x_1 x_3 - x_3(x_2 - 2x_4) \le 0$,

Variable range $10 \le x_1 \le 50$, $10 \le x_2 \le 80$, $0.9 \le x_3 \le 5$, $0.9 \le x_4 \le 5$.

References:

[1] Wolpert DH, Macready WG, “No free lunch theorems for optimization,” IEEE Trans Evol Comput, vol.1, pp.67-82, 1997.

[2] Holland JH, “Genetic algorithms,” Sci Am, vol.267, pp.66-72, 1992.

[3] R. C. Eberhart and J. A. Kennedy, “A new optimizer using particle swarm theory,” In Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS ’95), pp. 39-43, IEEE, Nagoya, Japan, October 1995.

[4] Aarts EHL, Laarhoven PJM, “Simulated annealing: an introduction,” Stat Neerl, vol.43, pp.31-52, 1989.

[5] Geem ZW, Kim JH, “A new heuristic optimization algorithm, harmony search,” Simulation, vol.76, pp.60-68, 2001.

[6] Rashedi E, Nezamabadi-Pour H, Saryazdi S, “GSA: a gravitational search algorithm,” Information Sciences, vol.179, no.13, pp.2232-2248, 2009.

[7] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” Knowledge-Based Systems, no.89, pp.228-249, 2015.

[8] S. Mirjalilia, “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Systems, no.27, pp.1-14, 2016.

[9] S. Mirjalilia, S. M. Mirjalili, A. Hatamlou, “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization,” Neural Computing and Applications, vol.17, pp.16-19, 2015.

[10] M. Tuba, I. Brajevic, R. Jovanovic, “Hybrid Seeker Optimization Algorithm for Global Optimization,” Applied Mathematics & Information Sciences, no. 3, pp.867-875, 2013.

[11] L. Yin, S. Luo, Y. Wang, et al., “Coordinated Complex-Valued Encoding Dragonfly Algorithm and Artificial Emotional Reinforcement Learning for Coordinated Secondary Voltage Control and Automatic Voltage Regulation in Multi-Generator Power Systems,” IEEE Access, vol. 8, pp.180520-180533, 2020.

[12] P. Wang, Y. Zhou, Q. Luo, et al., “Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey,” Neurocomputing, 407, pp.313-342, 2020.

[13] S. Zhang, Y. Zhou, Q. Luo, et al., “A Complex-Valued Encoding Satin Bowerbird Optimization Algorithm for Global Optimization,” Intelligent Computing Methodologies, Springer International Publishing AG, part of Springer Nature, August 2018.

[14] Y. Zhou, Z. Bao, Q. Luo, et al., “A Complex-valued Encoding Wind Driven Optimization with Greedy Strategy for 0-1 Knapsack Problem,” Applied Intelligence, vol. 46, pp.684-702, 2017.

[15] F. Miao, Y. Zhou, Q. Luo, “Complex-valued encoding symbiotic organisms search algorithm for global optimization,” Knowledge and Information Systems, vol. 1, pp.209-248, 2019.

[16] M. Abdel-Baset, H. Wu, Y.Zhou, “A complex encoding flower pollination algorithm for constrained engineering optimisation problems,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 8(2), pp.108-126, 2017.

[17] P. Wang, Y. Zhou, Q. Luo, et al., “Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey,” Neurocomputing, vol. 407, pp.313-342, 2020.

[18] C. Dai, Y. Zhu, W. Chen, “Seeker optimization algorithm,” In: Proc.2006 Inter. Conf. Computational Intelligence and Security, Guangzhou, China, IEEE Press, vol.1, pp.225-229, 2006.

[19] C. Dai, Y. Zhu, W. Chen, "Seeker Optimization Algorithm," Lecture Notes in Computer Science, vol.4456, pp.167-176, 2007.
[20] C. Dai, W. Chen, Y. Zhu, and X. Zhang, "Seeker Optimization Algorithm for Optimal Reactive Power Dispatch," IEEE Transactions on Power Systems, vol. 24, no. 3, pp.1218-1231, 2009.
[21] C. Dai, W. Chen, Y. Song, Y. Zhu, "Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization," Journal of Systems Engineering and Electronics, vol.21, no.2, pp.300-311, 2010.
[22] C. Dai, W. Chen, Y. Zhu, "Seeker optimization algorithm for digital IIR filter design," IEEE Transactions on Industrial Electronics, vol.57, no.5, pp.1710-1718, 2010.
[23] C. Dai, W. Chen, Y. Zhu, Z. Jiang, Z. You, "Seeker optimization algorithm for tuning the structure and parameters of neural networks," Neurocomputing, vol.74, no.6, pp.876-883, 2011.
[24] C. Dai, Z. Cheng, Q. Li, Z. Jiang, J. Jia, "Seeker optimization algorithm for global optimization: A case study on optimal modelling of proton exchange membrane fuel cell (PEMFC)," International Journal of Electrical Power and Energy Systems, vol.33, no.3, pp.369-376, 2011.
[25] C. Dai, W. Chen, L. Ran, Y. Zhang, Y. Du, "Human Group Optimizer with Local Search," Lecture Notes in Computer Science, vol.6728, pp.310-320, 2011.
[26] Y. Zhu, C. Dai, and W. Chen, "Seeker Optimization Algorithm for Several Practical Applications," International Journal of Computational Intelligence Systems, Vol. 7, No. 2, pp.353-359, 2014.
[27] D. Chen, H. Li, Z. Li, "Particle swarm optimization based on complex-valued encoding and application in function optimization," Comput, vol. 45, no. 10, pp.59-61, 2009.
[28] Li L, Zhou Y, "A novel complex-valued bat algorithm," Neural Computing and Applications, vol. 25, no. 6, pp.1369-1381, 2014.
[29] Y. Zhou, L. Li, M. Ma, "A complex-valued encoding bat algorithm for solving 0-1 knapsack problem," Neural Processing Letters, pp.1-24, 2015.
[30] Z. Zheng, Y. Zhang, Y. Qiu, "Genetic algorithm based on complex-valued encoding," IET Control Theory, pp.1-21, 2003.
[31] J. Kennedy, "Particle swarm optimization in Encyclopedia of Machine Learning," pp. 760-766, Springer, New York, NY, USA, 2010.
[32] H. Yu, H. Fang, P. Yao, et al., "A combined genetic algorithm/simulated annealing algorithm for large scale system energy integration," Computers & Chemical Engineering, vol. 24, no.8, pp.2023-2035, 2000.
[33] L. Li, Y. Zhou, and J. Xie, "A free search krill herd algorithm for functions optimization," Mathematical Problems in Engineering, vol. 2014, pp:1-21, 2014.
[34] X. Li, J. Zhang, and M. Yin, "Animal migration optimization: an optimization algorithm inspired by animal migration behavior," Neural Computing and Applications, vol. 24, no.7-8, pp.1867-1877, 2014.
[35] Molga M, Smutnicki C, "Test functions for optimization needs," http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf, 2005.
[36] Dasgupta S, Papadimitriou CH, Vazirani UV. Algorithms. McGraw-Hill, Inc. Professional Book Group 11 West 19th Street New York, NYUnited States: China Machine Press; pp.15-18, 2009.
[37] E. Dolan, J. Moré, "Benchmarking optimization software with performance profiles," Math. Program, vol.91 no.2, pp.201-213, 2002.
[38] Krohling RA, dos Santos Coelho L, "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems," IEEE Trans Syst Man Cybern Part B Cybern, vol.36, pp.1407-1416, 2006.
[39] Lee KS, Geem ZW, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," Comput Methods Appl Mech Eng, vol.194, pp.3902-3933, 2005.
[40] E. Mezura-Montes, C.A.C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," Int. J. Gen.Syst., vol.37, pp.443-473, 2008.
[41] L. Li, Z. Huang, F. Liu, Q. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," Comput. Struct, vol.85, pp.340-349, 2007.
[42] A. Kaveh, S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," Eng. Comput.: Int. J. Comput. Aid. Eng., vol.27, pp.155-182,2010.
[43] Deb K, "Optimal design of a welded beam via genetic algorithms," AIAA J, vol.29, pp.2013-2015, 1991.
[44] A.H. Gandomi, X.-S. Yang, A.H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," Eng. Comput, no.29, pp.17-35, 2013.
[45] H. Chickermane, H. Gea, "Structural optimization using a new local approximation

method," Int. J. Numer. Methods Eng., vol.39, pp.829-846, 1996.
[46] M.-Y. Cheng, D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," Comput. Struct, vol.139, pp. 98-112, 2014.
[47] E. Sandgren, "Nonlinear Integer and Discrete Programming in Mechanical Design," ASME Journal Mechanical Design, vol.112, pp.223-229, 1990.
[48] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, "Mine blast algorithm: a new population-based algorithm for solving constrained engineering optimization problems," Appl. Soft Comput, vol.13, pp.2592-2612, 2013.
[49] M. Zhang, W. Luo, X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," Inf. Sci, vol.178, pp.3043-3074, 2008.
[50] G.G. Wang, "Adaptive response surface method using inherited latin hypercube design points," J. Mech. Des., vol.125, pp.210-220, 2003.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Shaomi Duan carried out the simulation and the optimization, and wrote the first draft of the paper. Huilong Luo provided financial support, managed the project and revised the paper. Haipeng Liu has organized and executed the experiments of Section 4; he was responsible for the Statistics.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The data used to support the fundings of this study are available from the corresponding author upon request.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper. Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy) Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself