Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1344667, 28 pages
https://doi.org/10.1155/2022/1344667

Research Article

An Elastic Collision Seeker Optimization Algorithm for Optimization Constrained Engineering Problems

Shaomi Duan, [1,2] **Huilong Luo**, [1] **and Haipeng Liu** [2]

[1] *Faculty of Civil Engineering and Mechanics, Kunming University of Science and Technology, Kunming 650500, China*
[2] *Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China*

Correspondence should be addressed to Huilong Luo; huilongluo@kmust.edu.cn

Received 3 July 2021; Revised 23 September 2021; Accepted 11 December 2021; Published 10 January 2022

Academic Editor: Aditya Rio Prabowo

To improve the seeker optimization algorithm (SOA), an elastic collision seeker optimization algorithm (ECSOA) was proposed. The ECSOA evolves some individuals in three situations: completely elastic collision, completely inelastic collision, and non-completely elastic collision. These strategies enhance the individuals' diversity and avert falling into the local optimum. The ECSOA is compared with the particle swarm optimization (PSO), the simulated annealing and genetic algorithm (SA_GA), the gravitational search algorithm (GSA), the sine cosine algorithm (SCA), the multiverse optimizer (MVO), and the seeker optimization algorithm (SOA); then, fifteen benchmark functions, four PID control parameter models, and six constrained engineering optimization problems were selected for the experiment. According to the experimental results, the ECSOA can be used in the benchmark functions, the PID control parameter optimization, and the optimization constrained engineering problems. The optimization ability and robustness of ECSOA are better.

1. Introduction

Recently, the heuristic algorithm has received a lot of attention. Such algorithms create random methods for many optimization problems. Since the "no free lunch" (NFL) theorem, no one optimization solution can optimize overall questions [1]. Therefore, researchers pose new algorithms or enhance the current algorithms to deal with optimization problems. The current algorithms are the genetic algorithm (GA) [2], the particle swarm optimization (PSO) [3], the simulated annealing (SA) [4], the harmony search (HS) [5], the gravitational search algorithm (GSA) [6], the moth-flame optimization (MFO) [7], the sine cosine algorithm (SCA) [8], the multiverse optimizer (MVO) [9], the seeker optimization algorithm (SOA) [10], the artificial bee colony (ABC) algorithm [11], the krill herd (KH) [12], the monarch butterfly optimization (MBO) [13], the elephant herding optimization (EHO) [14], the moth search (MS) algorithm [15], the slime mould algorithm (SMA) [16], and the Harris hawks optimization (HHO) [17].

However, some optimization algorithms are still not very successful in optimization problems. The optimization problems include issues with low optimization precision, being premature, having only a local optimal solution, slow convergence speed, and insufficient robustness. To better overcome the issues of optimization precision, prematurity, having only a local optimal solution, slow convergence rate, and poor robustness, some improved algorithms have proven to be feasible optimization algorithms and have been used in practical engineering. For instance, the Harris hawks optimization algorithm, salp swarm algorithm, grasshopper optimization algorithm, and dragonfly algorithm are used for the structural design optimization of vehicle components [18]. The adaptive inertia weight factor in the traditional PSO optimizes path planning [19]. The PSO based on Gaussian and quantum behavior optimizes constrained engineering problems [20]. The least squares support vector machines based on Gaussian are proposed [21]. A Levy flights discrete bat algorithm is adopted to solve the Euclidean traveling salesman problem [22]. The cuckoo optimization algorithm in reverse logistics is used to design a network for COVID-19 waste management [23]. A chaotic cuckoo optimization algorithm based on a Levy flight, backward learning, and interfere operator is used to classify the optimal feature subspace [24]. An elite symbiotic organisms search algorithm with mutually beneficial factor is adopted to optimize the functions [25]. An artificial bee colony with dynamic Cauchy mutation is adopted to solve feature selection [26]. A new elastic collision optimization algorithm is applied in sensor cloud resource scheduling [27].

Dai et al. proposed the SOA in 2006 [28]; the goal is to mimic the seekers' behavior and the way they exchange information and solve practical application optimization problems. In the recent decade, the SOA has been used in many fields, such as unconstrained optimization problems [29], optimal reactive power dispatch [30], challenging set of benchmark problems [31], design of a digital filter [32], optimizing parameters of artificial neural networks [33], optimizing model and structures of fuel cell [34], novel human group optimizer algorithm [35], and several practical applications [36]. However, in the initial stage of dealing with optimization problems, SOA converges faster than others. When all individuals are near the best individual for solving the optimization problem, the individuals will lose diversity and fall into prematurity.

In this article, we propose an elastic collision seeker optimization algorithm (ECSOA), which evolves some individuals in three situations: complete elastic collision, complete inelastic collision, and incomplete elastic collision. These strategies enhance the individuals' diversity and avert premature convergence. The ECSOA is compared to seven improved SOAs, such as the changing algorithm parameters, the adaptive transformation of empirical value parameters, the Levy motion of some individuals, the reverse learning, the addition of mutual benefit factor, and the Cauchy mutation. This article chose fifteen benchmark functions to test. According to the experimental results, the convergence speed and accuracy of ECSOA are higher. The improved strategy enables the SOA to maintain the individuals' diversity, avert falling into the local optimum, and make up for the shortcomings of SOA in the aspect of easy precocity. Finally, compared with PSO, SA_GA, GSA, SCA, MVO, and SOA, the ECSOA has been implemented and tested on a complete set of well-known fifteen benchmark functions, four PID control parameter optimization models, and six optimization constrained engineering problems taken from literature. According to the experimental results, ECSOA is feasible in the benchmark functions, the PID parameter optimization problems, and the constrained engineering optimization problems. The ECSOA can find better values for solving the questions. The improved SOA successfully overcomes its tendency to prematurely converge to local optima for problems. The ECSOA has better optimization performance and robustness. The algorithm also has an improvement over the original SOA. The advantages of the ECSOA are summed up as follows:

(1) An ECSOA is raised to enhance the precision and robustness of the optimization process.

(2) The elastic collision strategies, the completely elastic collision, the completely inelastic collision, and the non-complete elastic collision, can improve the diversity of individuals, enhance local search, and avert premature convergence.

The rest of the article structure is as follows. Section 2 presents the SOA and the algorithm improvement strategies. Section 3 describes the ECSOA. Section 4 shows the algorithm optimization experiments, the results, and the analyses. Lastly, Section 5 gives some conclusions.

2. Basic SOA and Algorithm Improvement Strategies

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

2.1. Key Update Points for SOA. SOAs have three main updating steps.

2.1.1. 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.

$$\begin{aligned}\vec{f}_{i,\mathrm{e}}(t) &= \vec{p}_{i,\mathrm{best}} - \vec{x}_i(t),\\ \vec{f}_{i,a}(t) &= \vec{g}_{i,\mathrm{best}} - \vec{x}_i(t),\\ \vec{f}_{i,\mathrm{p}}(t) &= \vec{x}_i(t_1) - \vec{x}_i(t_2).\end{aligned} \tag{1}$$

The searcher uses the method of a random weighted average to obtain the search orientation.

$$\vec{f}_i(t) = \mathrm{sign}\left(\omega \vec{f}_{i,\mathrm{p}}(t) + \varphi_1 \vec{f}_{i,\mathrm{e}}(t) + \varphi_2 \vec{f}_{i,\mathrm{a}}(t)\right), \tag{2}$$

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 $\{\vec{x}_i(t-2), \vec{x}_i(t-1), \vec{x}_i(t)\}$ separately; $g_{i,\mathrm{best}}$ is the historical optimal location in the neighborhood where the ith search factor is located; $p_{i,\mathrm{best}}p$ is the optimal locality from the ith search factor to the current locality; ψ_1 and ψ_1 are random numbers in [0, 1]; and ω is the weight of inertia.

2.1.2. Search Step Size. The SOA refers to the reasoning of the fuzzy approximation ability. The SOA, through the computer language, describes some of the human natural languages that can simulate human intelligence reasoning search behavior. If the algorithm expresses a simple fuzzy rule, it adapts to the best approximation of the objective optimization problems. The greater search step length is more important. However, the smaller fitness corresponds to the smaller search step length. The Gaussian distribution function is adopted to describe the search step measurement.

$$\mu(\alpha) = e^{-\alpha^2/2\delta^2}, \tag{3}$$

where α and δ are parameters of a membership function.

According to (3), the probability of the output variable exceeding $[-3\delta, 3\delta]$ 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 an 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:

$$\mu_i = \mu_{\max} - \frac{s - I_i}{s - I}\left(\mu_{\max} - \mu_{\min}\right), \quad i = 1, 2, \cdots, s, \tag{4}$$

$$\mu_{ij} = \mathrm{rand}\left(\mu_i, 1\right), \quad j = 1, 2, \cdots, D, \tag{5}$$

where μ_{ij} is determined by (4) and (5), 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(μ_i,1) is the real number in any partition [μ_{i}, 1].

It can be seen that (4) simulates the random search behavior of human beings. Step measurement of j-dimensional search interspace is determined by the following equation:

$$\alpha_{\mathrm{ij}} = \delta_{ij} - \sqrt{-\ln\left(\mu_{ij}\right)}, \tag{6}$$

where δ_{ij} is a parameter of the Gaussian distribution function, which is defined by

$$\omega = \frac{\left(iter_{\max} - t\right)}{iter_{\max}}, \tag{7}$$

$$\delta_{ij} = \omega * \mathrm{abs}\left(\vec{x}_{\min} - \vec{x}_{\max}\right), \tag{8}$$

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.

2.1.3. Individual Location Updates. After obtaining the scout direction and scout step measurement of the individual, the location update is represented by (8).

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) f_{ij}(t), i = 1, 2, \cdots, s; j = 1, 2, \cdots, D. \tag{9}$$

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

2.2. Algorithm Improvement Strategies. Six strategies for improving the algorithm are listed in this paper.

2.2.1. Dynamic Adaptive Gaussian Variation of Empirical Parameters. In the SOA, (8) is changed to (10), and the empirical value C_1 is changed to an adaptive empirical value

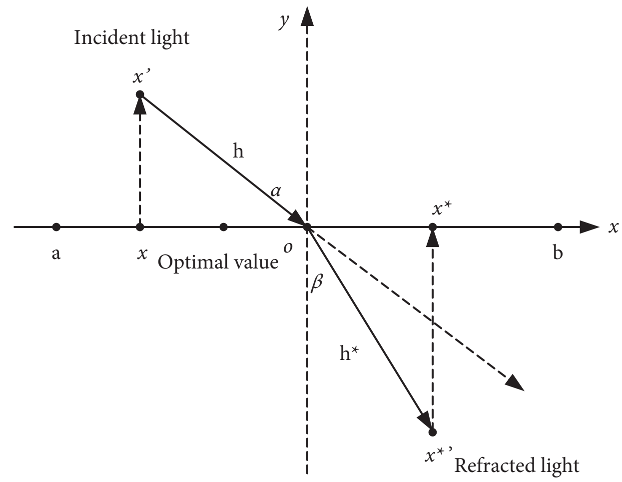

FIGURE 1: The schematic diagram of refraction reverse learning.

that varies between 0.1 and 0.5 with the change of optimization algebra according to (11). The individual position update is still the same as (9).

$$\delta_{\mathrm{ij}} = \omega * \mathrm{abs}\left(\vec{x}_{\min} - C_1 * \mathrm{rand}(1, d)\right), \tag{10}$$

$$C_1 = 0.5 - t * \left(\frac{0.1}{\mathrm{iter}_{\max}}\right), \tag{11}$$

where i represents the ith individual, j represents the individual dimension, δ_{ij} is a parameter of the Gaussian membership function [20, 21], t means the current algebra, $\mathrm{iter}_{\max}$ represents the maximum optimization algebra, and d represents the dimension of the optimized object.

2.2.2. The Levy Movement. A Levy movement [22, 24] is a random searching path alternating between short and occasionally long walks following the Levy distribution. The position update equation of the Levy motion is as follows:

$$\begin{aligned} x_{ij}(t+1) &= x_{ij}(t) \times \mathrm{Levy}(d), \\ \mathrm{Levy}(\beta) &= 0.01 \times \frac{\left(r_1 \times \sigma\right)}{\left|r_2\right|^{\beta^{-1}}}, \\ \sigma &= \left(\frac{\left(\Gamma(1+\beta) \times \sin(\pi\beta/2)\right)}{\left(\Gamma(1+\beta/2) \times \beta \times 2^{(\beta-1/2)}\right)}\right)^{\beta^{-1}}. \end{aligned} \tag{12}$$

i represents the ith individual and j represents the number of individuals. $\Gamma(\beta) = (\beta - 1)!$. t is the current algebra. d is the dimension of the optimized object. $r_1, r_2 \in \mathrm{rand}\,(0, 1)$. β is the partial real constant, which is 1.5 in this paper. After judging whether the fitness value is good or bad based on the newly generated individual position vector in (14), the original individual will be replaced by the best.

2.2.3. The Refraction Reverse Learning. If the projection of refraction points $x^{*\prime}$ on the x-axis represents x^*, x^* represent the reverse solution of individual x based on the refraction principle [23, 24]. The value of the boundary point in the refraction reverse learning is $(a + b)/2$ of the search interval [a, b]. As shown in Figure 1, the calculation of sinα and sinβ is shown in (13) and (14).

$$\sin \alpha = \frac{((a+b)/2 - x)}{h}. \tag{13}$$

$$\sin \beta = \frac{(x^* - (a+b)/2)}{h^*}. \tag{14}$$

According to (13) and (14), we can get

$$\frac{\sin \alpha}{\sin \beta} = \left(\frac{((a+b)/2 - x)}{(x^* - (a+b)/2)}\right)\left(\frac{h}{h^*}\right). \tag{15}$$

Assuming $k = h/h^*$, we can write (15) as (16), and then (17), $n = 1$ and $k = 1$, can be simplified to (18).

$$\mathrm{kn} = \left(\frac{((a+b)/2 - x)}{(x^* - (a+b)/2)}\right). \tag{16}$$

$$x^* = \frac{(a+b)}{2} + \frac{(a+b)}{(2kn)} - \frac{x}{(kn)}. \tag{17}$$

$$x^* = \mathrm{a} + \mathrm{b} - x. \tag{18}$$

When it is applied to the SOA, the probability of mutation is 0.8. The individual positions are taken for the refraction reverse learning according to (19) to get the new individual positions. In the formula, i is the ith individual, and j is the individual dimension. After judging whether the fitness value is good or bad based on the newly generated individual position vector in (19), the original individual is replaced by the best one.

$$x_{ij}^*(t+1) = \frac{(a_j + b_j)}{2} + \frac{(a_j + b_j)}{(2kn)} - \frac{x_{ij}(t)}{(kn)}. \tag{19}$$

2.2.4. The Mutually Beneficial Factor. The individuals x_h were randomly selected, and x_m was determined by (20) to determine the mutually beneficial factor C [25].

$$C = \frac{(x_h + x_m)}{2}. \tag{20}$$

$$x_{ij}(t+1) = x_{ij}(t) \times \varphi \times \left(x_{g\mathrm{best}} - C \times R\right), \tag{21}$$

where i represents the ith individual, j represents the individual dimension, ψ represents a random number in (0,1), x_{gbest} represents the j-dimensional component of the current optimal position of the entire population, C is the mutual benefit factor, R is the benefit parameter, and 1 or 2 is randomly selected. After judging whether the fitness value is good or bad based on the newly generated individual position vector in (21), the original individual is replaced by the best one.

2.2.5. The Cauchy Variation. In this paper, the Cauchy inverse can mutate the population under certain probability. The Cauchy inverse function [26] is shown in (22). Referring to (22), we can write the new position of the individual as (23); that is, the new position of the individual is obtained by the Cauchy mutation. After judging whether the fitness value is good or bad based on the newly generated individual position vector in (23), the original individual is replaced by the best one.

$$F^{-1}(p; x_0, \gamma) = x_0 + \gamma \cdot \tan\left(\pi \cdot \left(p - \frac{1}{2}\right)\right). \tag{22}$$

$$x_{ij}(t+1) = x_{ij}(t) + r_1 \cdot \tan\left(\pi \cdot \left(r_2 - \frac{1}{2}\right)\right), \tag{23}$$

where F^{-1} is the Cauchy inverse function and r_1 and r_2 are random values within [0, 1].

2.2.6. Elastic Collision Variation. For the individual x_{ij} (x_{ij} is a solution distributed in the solution space of the optimization problem and can be abstractly represented as a unit mass object at a certain position in the space), $\delta = \{x'\}(x' \in P(t) \wedge x' \neq x_{ij})$; x_{ij} and x' move in each other's direction at the velocities $f(x_{ij})$ and $f(x')$, respectively; and x_{ij} and x' will collide at Δt, and then after Δt, x_i reaches the new position $x_{i,new}$. The derivation is as follows. For the complete elasticity (CE) collision, according to the law of conservation of momentum and energy [27],

$$\begin{aligned} &CE: \left.\begin{aligned} &mf(x) + mf(x') = mf(x_{\mathrm{new}}) + mf(x'_{\mathrm{new}}), \\ &\frac{1}{2}m[f(x)]^2 + \frac{1}{2}m[f(x')]^2 = \frac{1}{2}m[f(x_{\mathrm{new}})]^2 + \frac{1}{2}m[f(x'_{\mathrm{new}})]^2, \end{aligned}\right\}, \\ &x_{\mathrm{new}} = \mathrm{CE}(x + x') = x\frac{2f(x')}{f(x') + f(x)} + x'\frac{f(x) - f(x')}{f(x') + f(x)}. \end{aligned} \tag{24}$$

Similarly, for the complete inelastic (CI) collisions and the non-complete elastic (NCE) collisions,

$$\begin{aligned} &CI\colon mf(x) + mf(x') = 2mf(x_{new}),\\ &x_{\text{new}} = \mathrm{CI}(x + x') = x\frac{3f(x') - f(x)}{2(f(x') + f(x))} + x'\frac{3f(x) - f(x')}{2(f(x') + f(x))},\\ &\text{NCE: } x_{\text{new}} = \mathrm{NCE}(x + x') = \beta\mathrm{CE}(x + x'); \beta \in (0, 1). \end{aligned} \tag{25}$$

The individual updating mechanism is as follows:

$$x_{ij}(t+1) = \begin{cases} \mathrm{ECO}\big(x_{ij}(t), \delta\big), & r_1 \geq \varepsilon,\\ x_{ij}(t), & r_1 < \varepsilon, \end{cases} \tag{26}$$

$$\mathrm{ECO}\big(x_{ij}(t), \delta\big) = \begin{cases} \mathrm{CE}\big(x_{ij}(t), \mathbf{G}_{ij}(t)\big), & r_2 \leq 0.5,\\ \mathrm{CI}\big(x_{ij}(t), \mathbf{B}_{ij}(t)\big), & 0.5 < r_2 \leq 0.5 + \xi,\\ \alpha\big[\mathrm{CE}\big(x_{ij}(t), \mathbf{G}_{ij}(t)\big) + \mathrm{CI}\big(x_{ij}(t), \mathbf{B}_{ij}(t)\big)\big], & \end{cases} \tag{27}$$

where i represents the ith individual, j represents the individual dimension, $\varepsilon \in (0,1)$ and $\xi \in (0,\ 0.5)$ are newer systems, $\boldsymbol{G}_{ij}(t)$ is the optimal solution of $x_{ij}(t)$ history, and $\boldsymbol{B}_{ij}(t)$ is the optimal solution of species. $r_1 \in (0, 1)$, $r_2 \in (0, 1)$, and $\alpha \in (0,\ 0.5)$ are the random numbers. After judging whether the fitness value is good or bad based on the newly generated individual position vector in (27), the original individual is replaced by the best one.

3. ECSOA

The ECSOA evolves some individuals in the CE collision, the CI collision, and the NCE collision to improve the diversity of individuals and boost partial scouting. Algorithm 1 is the primary process of the ECSOA.

4. Experimental Results

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

4.2. Algorithm Performance Comparison in Benchmark Functions. To ensure that the comparison of these algorithms is fair, the population number of algorithms 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.

4.2.1. Benchmark Functions. In this field, it is common to base the capability of algorithms on mathematic functions that are known to be globally optimal. Fifteen benchmark functions in the literature are used as the comparative test platform [7, 10, 37–39]. Table 1 shows the functions in the experiment. Variables are set to one hundred.

4.2.2. Performance Comparison of SOA with Different Improvement Methods. In this paper, the SOA is improved by seven different methods: the parameter changing SOA (PCSOA), the parameter adaptive Gaussian transform SOA (PAGTSOA), the SOA based on Levy variation (LVSOA), the SOA based on refraction reverse learning mechanism (RRLSOA), the SOA based on mutually beneficial factor strategy (MBFSOA), the SOA based on Cauchy variation (CVSOA), and the elastic collision seeker optimization algorithm (ECSOA).

(1) Parameter Setting of SOA with Different Improvement Methods. This section will introduce the parameter setting of the improved SOAs used in the experiment in this paper. Dai et al. have done a lot of research on the parameter set of the SOA [32], and we did a lot of practice tests and comparative studies about the parameters. The specific parameters of the improved SOA are shown in Table 2. In the next section, we will use these improved algorithms for experimental comparison and choose a relatively optimal improved algorithm to compare it with other advanced intelligent algorithms.

(2) Improved Algorithms' Performance Comparison in Benchmark Functions. The SOA is improved in seven different ways: the SOA based on parameter change (PCSOA),

(1) $t = 0$
(2) Parameter initialization.
(3) Population initialization. Generate an initial species group.
(4) Evaluate each seeker. Compute the fitness. Determine the optimal solution $\mathbf{P}_{\text{best,G}}$.
(5) While the stopping condition is not satisfied.
(5.1)Running process of the ECSOA
(1) The search direction of the searcher is generated according to (2)
(2) The search step size is generated according to (6)
(3) Generate a new position $x_{\text{ECSOA,G}}$ according to (9), and the range of $x_{\text{ECSOA,G}}$ is judged and modified to meet $(x_{\min}, x_{\max})$.
(4) Calculate the fitness and judge the optimal solution.
if $f(x_{\text{ECSOA,G}}) \leq \mathbf{P}_{\text{best,G}}$
$\mathbf{P}_{\text{best,G}} = f(x_{\text{ECSOA,G}})$
end if
(5.2) The elastic collision variation
(1) if rand $< P_m$, the elastic collision variation was carried out on some new positions, according to (26), to obtain new $x_{\text{ECSOA,G}}$, and the range of $x_{\text{ECSOA,G}}$ is judged and modified to meet $(x_{\min}, x_{\max})$.
(Other improvement strategies, such as the empirical value parameter adaptive transformation formula (10), the refraction reverse learning formula (14), the Levy variation formula (19), the introduction of mutually beneficial factor formula (21), and the Cauchy variation formula (23), were updated according to the corresponding formula.)
(2) Calculate the fitness and judge the optimal solution.
if $f(x_{\text{ECSOA,G}}) \leq \mathbf{P}_{\text{best,G}}$
$\mathbf{P}_{\text{best,G}} = f(x_{\text{ECSOA,G}})$
end if
end if
(6) $t = t + 1$
(7) if $t < T_{\max}$, then jump to 3; else stop.

Algorithm 1: ECSOA.

the SOA based on parameter adaptive Gaussian transform (PAGTSOA), the SOA based on Levy variation (LVSOA), the SOA based on refraction reverse learning mechanism (RRLOOA), the SOA based on mutual benefit factor strategy (MBFSOA), the SOA based on Cauchy variation (CVSOA), and the SOA based on elastic collision (ECSOA). To test the performance, each improved algorithm was optimized for the fifteen functions in Table 1. Each algorithm and each function were run independently 30 times. The performance of the SOA and the seven improved SOAs in fifteen-function optimization was compared in terms of the mean (Mean), standard deviation (Std.), best fitness (Best), program running time (Time), and best fitness rank (Rank) of 30 running results. The optimal fitness reflects the optimization accuracy of the algorithm, the average value and standard deviation reflect the robustness of the algorithms, and the running time reflects the time of the program. The results of the functions f1-f15 are displayed in Table 3. The boldface indicates that the optimal result is better.

Based on Table 3, for the benchmark functions f1-f15, the comparison between the seven improved SOAs in this paper and the original SOA shows that the optimization result of the ECSOA is the best value. The mean (Mean), standard deviation (Std.), best fitness (Best), and best fitness rank (Rank) of the ECSOA were the best after 30 independent runs. The f1-f15 total program running time (Time) rank is the fourth among all the eight algorithms compared in this paper. The running time of the ECSOA is longer than that of the SOA, PCSOA, and PAGTSOA; it is shorter than that of the LVSOA, MBFSOA, CVSOA, and PAGTSOA. From the perspective of optimization accuracy and robustness, the ECSOA has the best optimization performance among the improved SOAs in this paper. Section 4.2.3 will compare the ECSOA with the other intelligent optimization algorithms that are widely used at present.

4.2.3. Performance Comparison of Different Algorithms in Benchmark Functions. To test the performance of the ECSOA, it is compared to the PSO, SA_GA, GSA, SCA, MVO, and SSA, using the fifteen benchmark functions [7, 10, 37–39] in Table 1, which have been widely used in the test.

(1) The Parameter Setting of Different Algorithms. In this section, the parameters' set of the PSO [40], SA_GA [41], GSA [6], SCA [8], MVO [9], SOA [28], and ECSOA is presented. According to [6, 8, 9, 28, 40, 41], we did a lot of practice tests and comparative studies for the parameters set. Table 4 shows the parameters set of different algorithms.

(2) The Results Comparison of Different Algorithms in Benchmark Functions. The mean values, standard deviation, best fitness, and best fitness rank of the algorithms of 30 independent runs and the data of functions f1-f15 optimization results are shown in Table 5. The boldface indicates that the optimal outcome is better.

Based on Table 5, for the best value of the benchmark functions, the standard deviation, and the mean, the ECSOA is better than the others. According to the optimal fitness

TABLE 1: Description of benchmark functions.

Name	Test functions	Search	Minimum
Sphere	$f_1(x) = \sum_{i-1}^{n} x_i^2$	[−100,100]	0
Schwefel 2.22	$f_2(x) = \sum_{i-1}^{n} \lvert x_i \rvert + \prod_{i-1}^{n} \lvert x_i \rvert$	[−10,10]	0
Rotated hyperellipsoid	$f_3(x) = \sum_{i-1}^{n} \sum_{i-1}^{i} x_j^2$	[−100,100]	0
Schwefel 2.21	$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[−100,100]	0
Rosenbrock	$f_5(x) = \sum_{i-1}^{n-1} [100(x_{i+1} - x_{\infty}^2)^2 + (x_i - 1)^2]$	[−30,30]	0
Step	$f_6(x) = \sum_{i-1}^{n} (\lvert x_i + 0.5 \rvert)^2$	[−100,100]	0
Quartic	$f_7(x) = \sum_{i-1}^{n} i x_i^4 + \mathrm{random}(0,1]$	[−1.28,1.28]	0
SumSquares	$f_8(x) = \sum_{i-1}^{n} i x_i^2$	[−10,10]	0
SumPower	$f_9(x) = \sum_{i-1}^{n} \lvert x_i \rvert^{(i+1)}$	[−1.28,1.28]	0
Schwefel	$f_{10}(x) = -\sum_{i-1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	[−500,500]	−418.9829∗100
Rastrigin	$f_{11}(x) = \sum_{i-1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	0
Ackley	$f_{12}(x) = 20 + e - 20e^{-0.2}\sqrt{1/n \sum_{i-1}^{n} x_i^2} - e^{1/n} \sum_{i-1}^{n} \cos(2\pi x_i)$	[−32,32]	0
Griewank	$f_{13}(x) = 1/4000 \sum_{i-1}^{n} x_i^2 - \prod_{i-1}^{n} \cos(x_i/\sqrt{i}) + 1$	[−600,600]	0
Penalized1	$\mathrm{f}_{14}(x) = \frac{\pi}{\mathrm{n}} \left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (\mathrm{y_n} - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	$y_i = 1 + 1/4(x_i + 1), u(x_{i,a,k,m}) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & a \le x_i \le a \\ (-x_i - a)^m, & x_i < -a \end{cases}$	[−50,50]
Penalized2	$\mathrm{f}_{15}(x) = \frac{1}{10} \left\{\sin^2(3\pi \mathrm{x}_1) + \sum_{i=1}^{\mathrm{n}-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (\mathrm{x_n} - 1)^2 [1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{\mathrm{n}} u(x_i, 5, 100, 4)$	[−50,50]	0

Table 2: Parameter settings of SOA with different improvement methods.

Algorithm	Parameters and values
SOA [5]	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$.
PCSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$.
PAGTSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.1$–0.5.
LVSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$, the Levy mutation probability: $P_L=0.8$.
RRLSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$, the refraction reverse probability: $P_r=0.8$.
MBFSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$, the mutually beneficial factor probability: $P_m=0.8$.
CVSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$, the Cauchy variation probability: $P_C=0.8$.
ECSOA	The maximum membership degree value: $U_{\max}=0.95$, the minimum membership degree value: $U_{\min}=0.0111$, the maximum inertia weight value: $W_{\max}=0.9$, the minimum inertia weight value: $W_{\min}=0.1$, the empirical value: $W=0.2$, the elastic collision probability: $P_e=0.8$.

Table 3: Performance comparison of different SOA improvement strategies of 30 independent runs for benchmark functions.

Function	Result	Algorithms							
		SOA	PAGTSOA	PASOA	LVSOA	RRLSOA	MBFSOA	CVSOA	ECSOA
f_1 $(D=100)$	Mean	1.0524957	10.651095	5.8792970	0.1969439	0.0540391	1.4202532	3.5095128	**0**
	Std.	0.2051906	49.006900	26.483115	0.1757506	0.0941245	4.0124150	12.582150	**0**
	Best	0.6215967	0.0194690	0.0126392	9.264e-04	9.538e-06	0.0075097	0.0037232	**0**
	Time	**45.50061**	55.33285	53.03482	72.312269	138.09713	53.66496	48.995307	63.00413
	Rank	8	7	6	3	2	5	4	**1**
f_2 $(D=100)$	Mean	13.002085	0.8551306	1.1247211	0.4843834	0.0192347	0.8807482	0.8554324	**0**
	Std.	2.0008395	0.3078019	0.1241726	0.0896276	0.0033951	0.2147558	0.2150637	**0**
	Best	9.4729280	0.3884975	0.8443990	0.3522036	0.0139691	0.5532238	0.5829066	**0**
	Time	54.16754	51.23705	**49.17031**	72.312269	138.09713	53.66496	48.995307	68.65450
	Rank	8	4	7	3	2	5	6	**1**
f_3 $(D=100)$	Mean	9.869e+03	3.392e+03	3.452e+03	2.524e+03	4.8034565	4.346e+03	2.945e+03	**0**
	Std.	3.472e+03	1.658e+03	1.959e+03	2.193e+03	20.624508	2.943e+03	2.83e+03	**0**
	Best	3.318e+03	4.645e+02	37.813122	7.6866138	3.279e-04	2.408e+02	2.2538366	**0**
	Time	423.27283	**393.18358**	468.54936	639.09729	802.52870	519.93894	**375.72695**	720.86686
	Rank	8	7	5	4	2	6	3	**1**
f_4 $(D=100)$	Mean	21.743821	18.412017	18.418584	16.663924	0.0344293	19.610919	16.210237	**0**
	Std.	5.6809434	6.8890641	6.6275327	5.7602336	0.1376098	6.0681946	7.5363261	**0**
	Best	2.9045088	0.2149959	0.5369130	0.4595301	7.323e-04	2.5876468	0.3657577	**0**
	Time	**47.98446**	55.85934	49.86405	74.121127	101.07014	90.23634	49.264932	67.91751
	Rank	8	3	6	5	2	7	4	**1**
f_5 $(D=100)$	Mean	7.093e+02	1.334e+02	1.082e+02	98.378869	**97.910630**	3.843e+03	1.120e+02	98.124573
	Std.	1.675e+02	1.489e+02	14.576662	1.9535375	1.1793911	1.792e+04	1.284e+02	**0.0339669**
	Best	2.789e+02	96.133568	97.022113	95.965118	95.896077	96.241454	**16.545566**	98.057258
	Time	**54.515291**	62.169550	50.996834	122.32891	102.59963	68.034444	63.561199	79.987520
	Rank	8	4	6	3	2	5	1	**7**
f_6 $(D=100)$	Mean	1.1365348	10.241728	8.3207707	7.3932402	8.2483572	44.121188	3.4131678	**0.0471172**
	Std.	0.1751583	7.3068384	4.9997548	4.2448245	3.8398778	2.080e+02	6.5430302	**0.0339465**
	Best	0.7992740	0.2161040	0.0424130	0.2796657	0.0246273	**0.0166825**	0.0202905	0.0218316
	Time	**47.876927**	52.583180	46.059290	114.87808	94.008038	73.899842	48.275220	68.786422
	Rank	8	6	5	7	4	**1**	2	3

Table 3: Continued.

Function	Result	Algorithms							
		SOA	PAGTSOA	PASOA	LVSOA	RRLSOA	MBFSOA	CVSOA	ECSOA
f_7 ($D=100$)	Mean	3.4645595	0.0979540	0.1900503	0.0668954	3.532e-04	0.1034386	0.1007809	**9.621e-05**
	Std.	0.9443482	0.0295516	0.0493722	0.0198944	1.419e-04	0.0259960	0.0265230	**7.291e-05**
	Best	2.0472780	0.0497600	0.0989682	0.0386301	1.334e-04	0.0591166	0.0651808	**2.056e-05**
	Time	90.52067	95.46660	**85.11105**	171.49167	169.29825	109.08514	97.226073	162.01431
	Rank	8	4	7	3	2	5	6	1
f_8 ($D=100$)	Mean	1.391e+02	0.3830840	1.2523149	0.2155477	3.125e-04	0.4082059	0.4914759	**0**
	Std.	33.925547	0.1176738	0.2579516	0.0689183	9.806e-05	0.1798079	0.5899702	**0**
	Best	79.288776	0.2063376	0.7213791	0.1216883	1.766e-04	0.2088904	0.2163823	**0**
	Time	55.51168	51.83529	**47.67628**	113.84626	107.06077	62.40435	**48.359841**	68.06153
	Rank	8	4	7	3	2	5	6	**1**
f_9 ($D=100$)	Mean	1.471e-05	2.988e-08	1.597e-07	1.684e-08	1.683e-11	3.108e-08	4.428e-08	**0**
	Std.	1.279e-05	1.944e-08	1.098e-07	9.608e-09	1.695e-11	2.652e-08	4.378e-08	**0**
	Best	6.318e-07	4.835e-09	9.134e-09	3.497e-09	3.468e-13	5.062e-09	5.396e-09	**0**
	Time	102.88770	105.16369	**94.27801**	210.95845	262.27585	184.68808	142.54992	120.19844
	Rank	8	4	7	3	2	5	6	**1**
f_{10} ($D=100$)	Mean	−2.324e+4	−2.184e+4	−2.296e+4	−2.685e+4	−2.752e+4	**−4.151e+4**	−3.298e+4	−2.627e+4
	Std.	3.396e+03	3.105e+03	3.478e+03	5.409e+03	4.717e+03	**9.886e+02**	5.590e+03	7.262e+03
	Best	−3.174e+4	−2.906e+4	−3.383e+4	−3.942e+4	−3.658e+4	**−4.190e+4**	**−4.190e+4**	−4.105e+4
	Time	**60.474337**	64.865537	69.718244	134.61052	135.83717	96.863046	72.777288	82.973951
	Rank	7	8	6	4	5	**1**	**1**	3
f_{11} ($D=100$)	Mean	4.147e+02	40.408968	1.671e+02	16.244213	0.0046895	36.266581	18.815332	**0**
	Std.	47.133648	58.557280	52.557303	27.473872	0.0019906	46.628463	27.008433	**0**
	Best	3.230e+02	0.1867154	49.014907	0.2093659	0.0024715	0.1742839	0.2374936	**0**
	Time	59.59937	70.80121	**53.881313**	122.41571	215.67516	81.87741	66.201350	70.32757
	Rank	8	4	7	5	2	3	6	**1**
f_{12} ($D=100$)	Mean	2.4523846	0.4491129	0.2960575	0.3706855	0.0229150	0.5816069	0.2511530	**8.882e-16**
	Std.	0.3106584	0.6639778	0.5909855	0.9556260	0.0636082	1.1166917	0.6797442	**0**
	Best	1.9634230	0.0154282	0.0320302	0.0132288	9.955e-04	0.0177150	0.0136431	**8.882e-16**
	Time	64.89326	68.48468	**56.26708**	139.44284	188.30542	83.18719	57.217278	74.98492
	Rank	8	5	7	3	2	6	4	**1**
f_{13} ($D=100$)	Mean	0.5354222	0.9578562	0.143592	0.0806190	0.0344354	1.025111	0.9414278	**0**
	Std.	0.3343660	3.0280038	0.336573	0.2045818	0.1153465	2.874560	2.1213929	**0**
	Best	0.0806019	0.0020953	0.007806	0.0012407	3.238e-06	0.003087	0.0018399	**0**
	Time	63.67314	65.48267	**60.59472**	169.61463	179.53784	101.38439	65.805859	82.41825
	Rank	8	5	7	3	2	6	4	**1**
f_{14} ($D=100$)	Mean	20.231984	11.239536	13.286278	10.512085	0.0574088	14.582338	16.625816	**0.0355875**
	Std.	8.8688013	6.6059206	8.0516954	4.8185197	0.0487519	8.6789426	9.4591054	**0.0135230**
	Best	8.9451797	0.3752616	0.2490416	0.2575901	0.0211033	0.0101677	**0.0018590**	0.0020366
	Time	**165.94752**	214.22263	158.94344	264.67286	516.74399	246.38223	179.81703	211.78782
	Rank	8	7	5	6	4	3	**1**	2
f_{15} ($D=100$)	Mean	1.301e+02	1.080e+02	95.968154	45.734524	24.515453	2.014e+02	93.843578	**7.6774427**
	Std.	62.979168	78.512763	75.121239	49.570748	36.817491	6.687e+02	83.666000	**4.1034906**
	Best	1.7998261	9.9178264	9.7032945	9.3364901	5.1884504	9.8278162	0.2503571	**0.0977197**
	Time	**166.65126**	164.31928	159.80885	245.52676	484.10688	191.20796	211.63275	207.40591
	Rank	3	8	6	5	4	7	2	**1**
Average rank		7.6	5.33333	6.26667	4	2.6	4.66667	3.73333	1.73333
Overall rank		8	6	7	4	2	5	3	**1**

value mean rank and all rank results from Table 5, the ECSOA has a strong optimization ability and strong robustness to a benchmark function.

Figure 2 shows the fitness curves of the best values for the benchmark functions f1-f15 ($D=100$). As seen from Figure 2, the convergence of the ECSOA is faster, and the precision of the ECSOA is better.

Figure 3 is the ANOVA for the benchmark functions f1–f15 ($D=100$). As seen from Figure 3, the ECSOA showed better robustness and improved SOA. Therefore, the ECSOA is a feasible solution in the optimization of benchmark functions.

4.2.4. Complexity Analysis. The calculational complexity of the SOA is **O** (NDM), N represents the total individual count, D represents the dimension count, and M represents the maximum count of algebras. The computational

Table 4: The parameters set of different algorithms.

Algorithm	Parameters and values
PSO [40]	Constant inertia: $c0 = 0.9\sim0.4$, two acceleration coefficients: $c1 = c2 = 1.4962$.
SA_GA [41]	Selection probability: $P_s = 0.6$, crossover probability: $P_c = 0.7$, mutation scale factor: $P_m = 0.05$, initial temperature: $T_0 = 100$, temperature reduction parameter: $P_t = 0.98$.
GSA [6]	The gravitational constant: $G0 = 100$, alfa = 20.
SCA [8]	The random numbers: $r1 = 0\sim2$, $r2 = 0\sim2\pi$, $r3 = 0\sim2$, $r4 = 0\sim1$.
MVO [9]	The wormhole existence probability: WEP_Max = 1, WEP_Min = 0.2, traveling distance rate: TDR = 0~1, the random numbers: $r1 = 0\sim1$, $r2 = 0\sim1$, $r3 = 0\sim1$.
SOA [28]	The maximum membership degree value: $U_{\max} = 0.95$, the minimum membership degree value: $U_{\min} = 0.0111$, the maximum inertia weight value: $W_{\max} = 0.9$, the minimum inertia weight value: $W_{\min} = 0.1$. (The same as Table 2.)
ECSOA	The maximum membership degree value: $U_{\max} = 0.95$, the minimum membership degree value: $U_{\min} = 0.0111$, the maximum inertia weight value: $W_{\max} = 0.9$, the minimum inertia weight value: $W_{\min} = 0.1$, the empirical value: $W = 0.2$, the elastic collision probability: $P_e = 0.8$. (The same as Table 2.)

Table 5: Performance comparison of algorithms for benchmark functions.

Test functions	Result	Algorithms						
		PSO	SA_GA	GSA	SCA	MVO	SOA	ECSOA
$f_1(D = 100)$	Mean	0.0031	3.018e+04	6.21 E+02	6.42 E+03	40.269355	1.0524957	**0**
	Std.	6.31E−04	6.691e+03	2.71 E+02	5.62 E+03	6.8812839	0.2051906	**0**
	Best	0.0022	1.708e+04	2.50 E+02	96.29169	27.3763	0.6216	**0**
	Rank	2	7	6	5	4	3	**1**
f_2 $(D = 100)$	Mean	0.2192801	118.6375	7.1387617	1.3795842	7.34 E+20	13.002085	**0**
	Std.	0.0260935	13.168469	3.2135126	1.4335667	3.92 E+21	2.0008395	**0**
	Best	0.1732089	92.513924	2.0654838	0.0310204	5.43 E+02	9.4729280	**0**
	Rank	3	7	4	2	6	5	**1**
f_3 $(D = 100)$	Mean	2.27 E+03	3.77e+05	8.80 E+03	1.84 E+05	4.24 E+04	9.87 E+03	**0**
	Std.	1.30 E+03	8.95e+04	1.84 E+03	3.48 E+04	6.09 E+03	3.47 E+03	**0**
	Best	8.43 E+02	2.55e+05	6.26 E+03	1.24 E+05	3.11 E+04	3.32 E+03	**0**
	Rank	2	7	4	6	5	3	**1**
f_4 $(D = 100)$	Mean	1.4624169	89.626880	15.671857	85.539422	49.557442	21.743821	**0**
	Std.	0.2035038	3.454423	1.5605923	3.5980134	6.2884298	5.6809434	**0**
	Best	1.0929345	83.496264	10.958917	77.909721	37.299715	2.9045088	**0**
	Rank	2	7	4	6	5	3	**1**
f_5 $(D = 100)$	Mean	2.515e+02	5.783e+07	1.784e+04	6.427e+07	2.440e+03	7.093e+02	**98.124573**
	Std.	59.018749	2.620e+07	1.578e+04	4.671e+07	1.959e+03	1.675e+02	**0.0339669**
	Best	1.236e+02	1.925e+07	1.987e+03	7.162e+06	7.755e+02	2.789e+02	**98.057258**
	Rank	2	7	5	6	4	3	**1**
f_6 $(D = 100)$	Mean	**0.0119147**	3.074e+04	6.995e+02	6.131e+03	40.0892130	1.1365348	0.0471172
	Std.	**0.0020782**	8.323e+03	3.890e+02	4.651e+03	5.9162672	0.1751583	0.0339465
	Best	**0.008661**	1.356e+04	2.169e+02	6.729e+02	28.9167207	0.799274	0.0218316
	Rank	1	7	5	6	4	3	2
f_7 $(D = 100)$	Mean	0.2211229	77.487242	2.3136998	73.024146	0.3355664	3.4645595	**9.621e−05**
	Std.	0.0419681	49.238470	1.1506156	46.772681	0.0733789	0.9443482	**7.291e−05**
	Best	0.1180966	19.170953	0.8371509	11.108763	0.2176315	2.0472780	**2.06e−05**
	Rank	2	7	4	6	3	5	**1**
f_8 $(D = 100)$	Mean	0.0624146	1.29e+04	1.32 E+02	1.52 E+03	1.05 E+02	1.39 E+02	**0**
	Std.	0.0164020	3.57e+03	74.414138	1.16 E+03	43.821833	33.925549	**0**
	Best	0.0291246	7.55e+03	31.981022	2.47 E+02	34.296414	79.288776	**0**
	Rank	2	7	3	6	4	5	**1**
f_9 $(D = 100)$	Mean	2.08E−26	1.78e+03	3.67E−12	35.077327	1.21E−06	1.47E−05	**0**
	Std.	3.99E−26	9.54e+03	7.66E−12	80.100469	5.23E−07	1.28E−05	**0**
	Best	2.07E−29	0.002883	4.23E−16	0.0268174	4.55E−07	6.32E−07	**0**
	Rank	2	6	3	7	4	5	**1**
f_{10} $(D = 100)$	Mean	−4.746e+3	−2.475e+04	−4.729e+3	−7.256e+3	−2.408e+4	−2.324e+4	**−2.627e+4**
	Std.	**6.889e+02**	8.845e+02	8.702e+02	6.962e+02	1.533e+03	3.396e+03	7.262e+03
	Best	−6.440e+3	−2.659e+04	−7.354e+3	−9.457e+3	−2.658e+4	−3.174e+4	**−4.105e+4**
	Rank	7	3	6	5	4	2	**1**

Table 5: Continued.

Test functions	Result	Algorithms						
		PSO	SA_GA	GSA	SCA	MVO	SOA	ECSOA
f_{11} ($D = 100$)	Mean	33.896488	425.8326	1.36 E+02	2.06 E+02	6.41 E+02	4.15 E+02	**0**
	Std.	6.2350851	48.908770	18.532649	92.249647	69.443553	47.133647	**0**
	Best	23.655297	3.34 E+02	88.161003	67.314606	5.01 E+02	3.30 E+02	**0**
	Rank	2	6	4	3	7	5	**1**
f_{12} ($D = 100$)	Mean	0.0220555	15.3479726	3.1378392	18.309269	6.5910720	2.4523845	**8.882*e*−16**
	Std.	0.0032932	0.7816167	0.6404753	4.7183569	6.0322718	0.3106583	**0**
	Best	0.0161767	13.7962876	2.0521342	6.872015	3.2031355	1.9634230	**8.882*e*−16**
	Rank	2	7	4	6	5	3	**1**
f_{13} ($D = 100$)	Mean	9.1115292	274.5795	98.761595	53.145123	1.3775687	0.5354221	**0**
	Std.	1.8646352	66.687333	11.527169	38.195663	0.0626040	0.3343659	**0**
	Best	6.1937260	137.5120	81.688324	1.7219609	1.2241138	0.0806019	**0**
	Rank	5	**7**	6	4	3	2	**1**
f_{14} ($D = 100$)	Mean	**0.0167129**	4.418*e*+07	4.5497727	1.582*e*+08	11.6567511	20.2319839	0.0355875
	Std.	0.0241246	5.110*e*+07	1.21396147	1.352*e*+08	4.2010589	8.8688013	**0.0135230**
	Best	**4.838*e*−05**	3.883*e*+06	2.1673113	6.757*e*+06	6.7785747	8.9451797	0.0020366
	Rank	1	6	3	7	4	5	2
f_{15} ($D = 100$)	Mean	**0.0010358**	1.435*e*+08	1.309*e*+02	2.674*e*+08	1.209*e*+02	1.301*e*+02	7.6774427
	Std.	**0.002800**	1.008*e*+08	64.254054	1.527*e*+08	30.717732	62.9791683	4.1034906
	Best	**1.898*e*−04**	2.309*e*+07	76.4103682	3.106*e*+07	51.2656139	1.7998261	0.0977197
	Rank	1	6	5	7	4	3	2
Average rank		2.4	6.466667	4.4	5.466667	4.4	3.666667	1.2
Overall rank		2	7	4	6	4	3	**1**

complexity of the first phase of the SOA stage is **O** (NDM). The elastic collision strategy is introduced to calculate the **O** (NDM) value. Therefore, the overall complexity of the ECSOA is **O** (NDM + NDM). Based on the principle of the Big-O representation [42], if the count of algebras is high (M ≫ N, D), the calculational complexity is **O** (NDM). Therefore, the overall calculational complexity of the ECSOA is almost the same as the basic SOA.

4.2.5. Statistical Testing of Algorithms in Benchmark Functions. Using Wilcoxon's rank-sum test [43], we can discover the important differences between the two algorithms. This test gives the value $p < 0.05$.

Table 6 indicates the results of statistical testing. N/A represents the best algorithm. From Table 6, ECSOA is suitable for the fifteen functions. Therefore, the ECSOA is better than the other algorithms.

4.2.6. Run Time Comparison of Algorithms in Benchmark Functions. In this subsection, the running time of the algorithms for each function is recorded under the same conditions: population number of 30, evolution algebra of 1000, and 30 independent runs of the above fifteen benchmark functions f1–f15 ($d = 100$). Then, the running time of the fifteen functions is added to obtain the sum of the 30 independent running times of each algorithm for the fifteen functions listed in this paper and the ranking of the total time, as shown in Table 7. As seen from Table 7, the SCA has the most minor program running time, followed by the PSO algorithm, which has more program running time. The ECSOA ranks fifth, which has a relatively longer program running time. At the bottom of the list is the SA_GA, which takes the most running time.

To learn more traits about the program running time of the seven algorithms in the fifteen functions, a bar chart in Figure 4 was made for the total time of each algorithm after 30 independent runs. From Figure 4, as to the running time, the ECSOA is less than the SA_GA and GSA; the SCA is the least; the SA_GA is the most; the ECSOA is less than one-sixth of SA_GA; and the ECSOA is nearly four times the SCA, which is relatively large.

4.2.7. 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 is calculated by the following formulas:

$$r_{f,g} = \frac{\mu_{f,g}}{\min\left\{\mu_{f,g}\colon g \in G\right\}}, \tag{28}$$

$$\rho_g(\tau) = \frac{size\left\{f \in F\colon r_{f,g} \le \tau\right\}}{n_f}, \tag{29}$$

where g represents an algorithm; **G** is the algorithms set; f means a function; **F** represents the function set; n_g represents the count of algorithms in the experiment; n_f is the number of functions in the experiment; $\mu_{f,g}$ is the average fitness obtained by the algorithm g after solving function f, $r_{f,g}$ is the capability ratio; ρ_g is the algorithmic capability; and τ is a factor of the best probability [44].

Figure 5 shows the capability ratios of the average value for the seven algorithms on the benchmark functions f1-f15 ($D = 100$). The consequences are revealed by a log scale 2. As

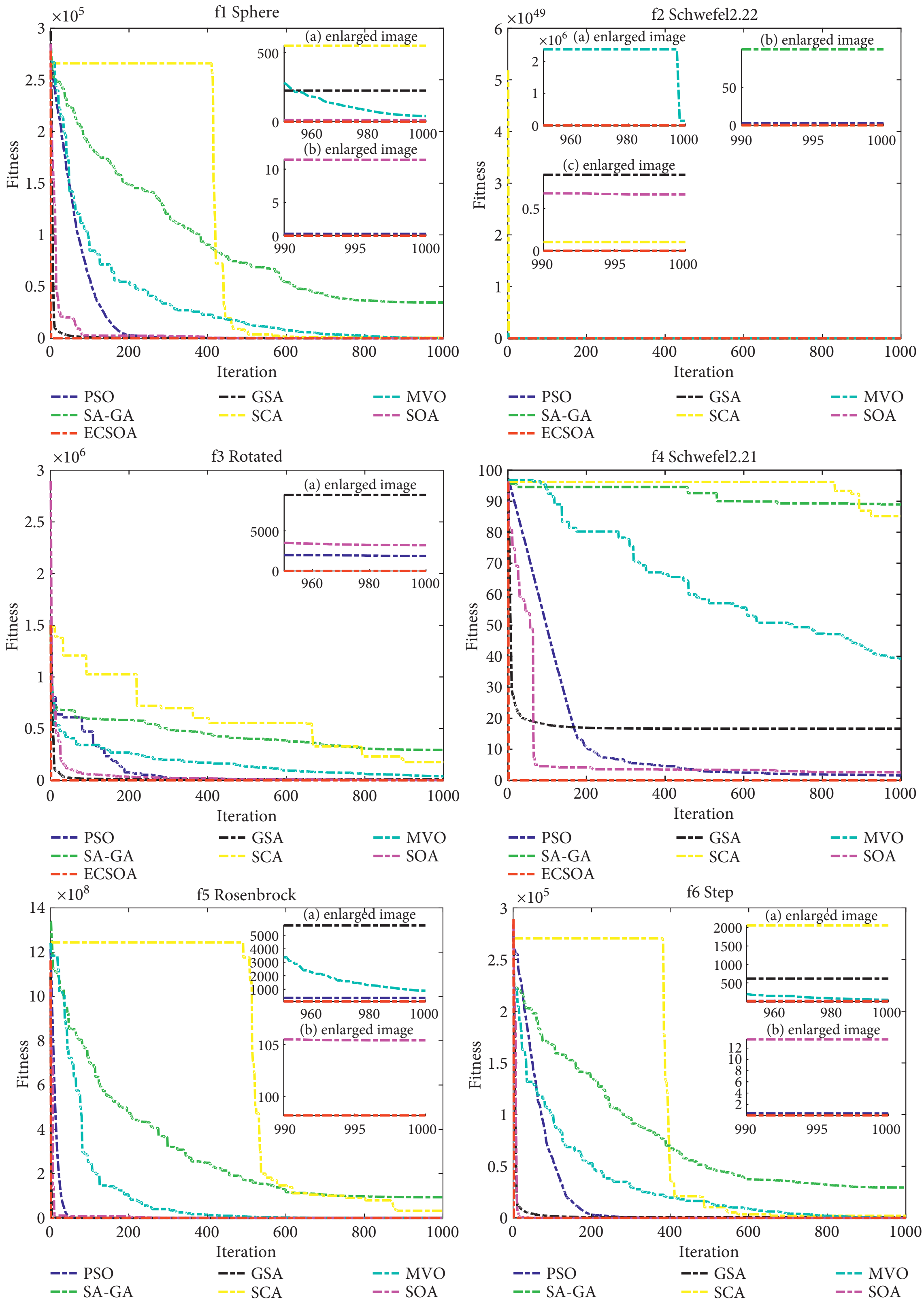

Figure 2: Continued.

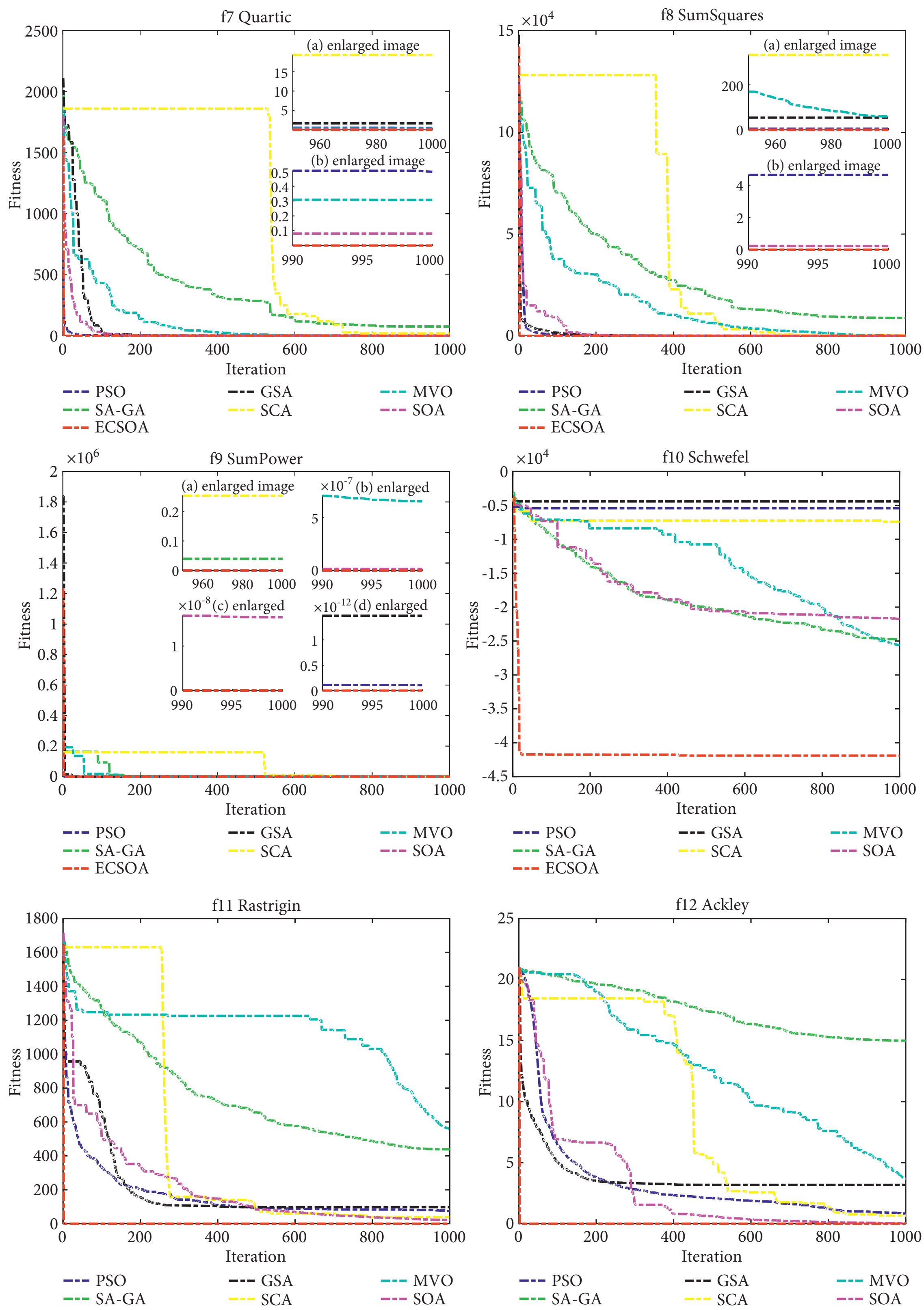

Figure 2: Continued.

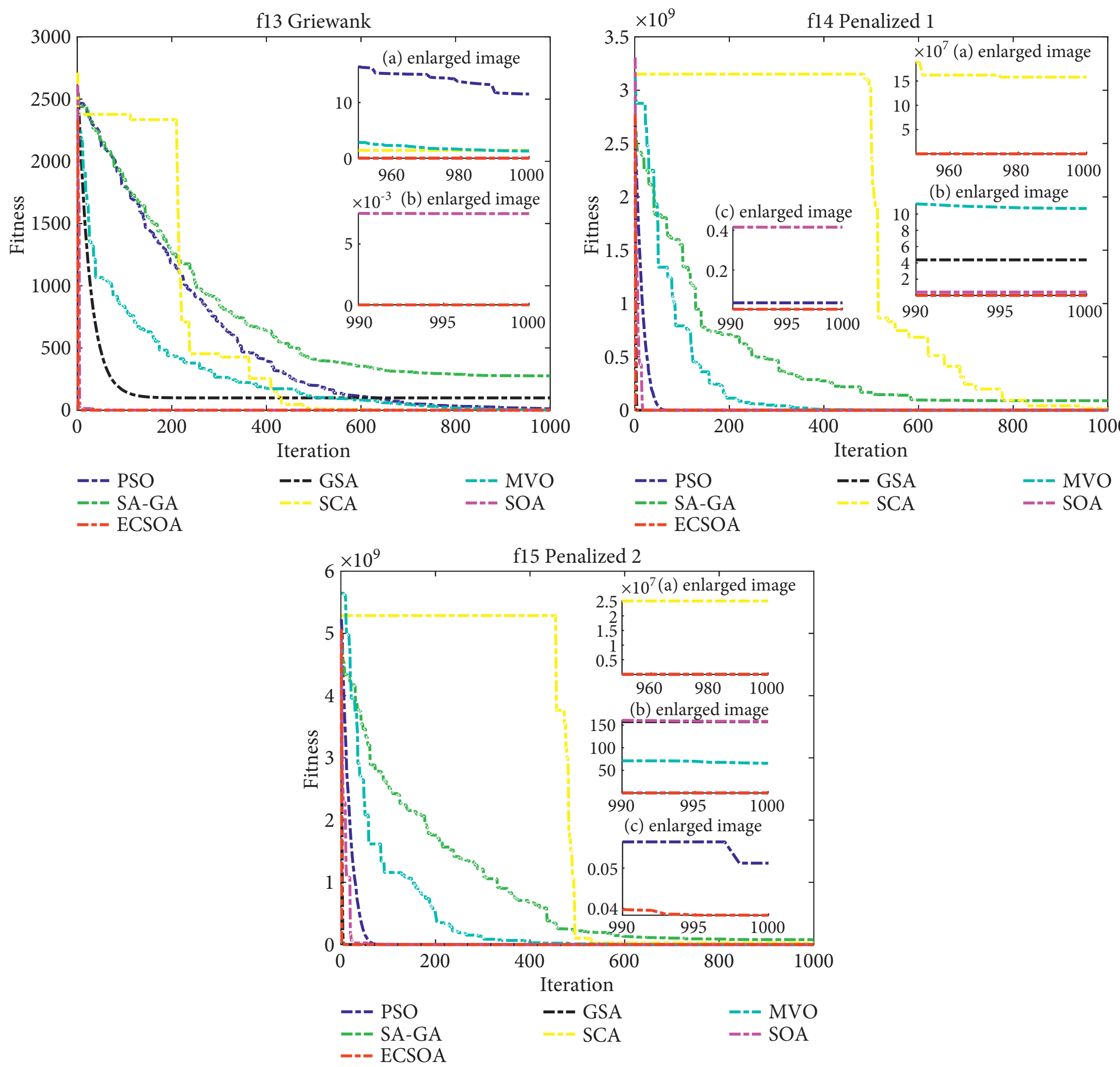

FIGURE 2: Convergence curves for benchmark functions f1–f15 ($D = 100$).

shown in Figure 5, the ECSOA has the highest probability. When $\tau = 1$, the ECSOA is about 0.8, which is better than that of the others. When $\tau = 8$, the ECSOA is the winner on the given test functions, ESOA is 1, PSO is 0.67, SA_GA is 0.067, SCA is 0.3, GSA is 0.3, MVO is 0.2, and SOA is 0.33. Regarding the performance curve, the ECSOA is the best; the ECSOA can achieve 100% when $\tau \geq 1$. Thus, the performance of the ECSOA is better than that of the other algorithms.

4.3. Algorithm Performance Comparison in PID Controller Parameter Optimization Problems. In this subsection, we use four control system optimizing PID parameter models to test the capability of the ECSOA. For g1–g3, the population number of all algorithms is 20, the max number of algebras is 20, g1-g2 step response time is set to 10s, and g3 step response time is set to 30s. For g4, the population number of all algorithms is 50, the max number of algebras is 50, the step response time is set to 50s.

4.3.1. Control System Models. Equations (30)–(33) show the test control system models optimizing PID parameters used in our experiment. Figure 6 shows the process diagram for optimizing the test control system PID parameters by the ECSOA. Figure 7 shows the optimization PID parameter model structure of the control system.

$$g_1(s) = \frac{2.6}{(2.7s + 1)(0.3s + 1)}, \tag{30}$$

$$g_2(s) = \frac{5}{(2.7s + 1)} e^{-0.5s}, \tag{31}$$

$$g_3(s) = \frac{3}{(2s + 1)} e^{-3s}, \tag{32}$$

$$g_4(s) = \frac{1}{(s + 1)^8}. \tag{33}$$

4.3.2. Results Comparison of Algorithms in the PID Controller Parameter Optimization. For testing the capability of the

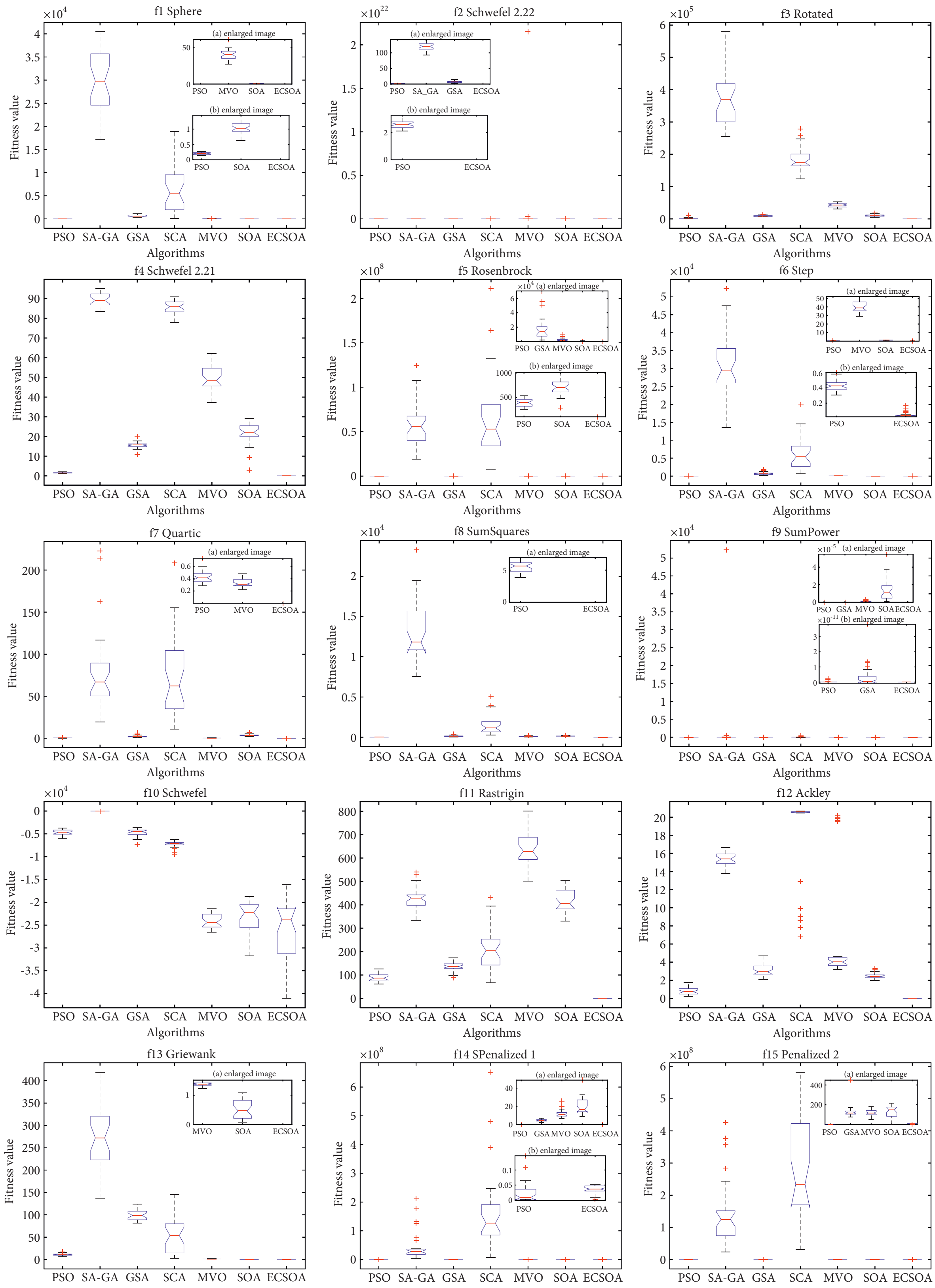

Figure 3: ANOVA tests for benchmark functions f1–f15 ($D = 100$).

Table 6: The p values of the Wilcoxon rank-sum test.

Test functions	p test values of various algorithms						
($D = 100$)	PSO	SA_GA	GSA	SCA	MVO	SOA	ECSOA
f_1	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_2	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_3	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_4	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_5	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	N/A
f_6	N/A	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11
f_7	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	N/A
f_8	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_9	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_{10}	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	0.695215399	0.129670225	N/A
f_{11}	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_{12}	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_{13}	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	1.21178e-12	N/A
f_{14}	N/A	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	8.11998e-04
f_{15}	N/A	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11	3.01986e-11

Table 7: Run time comparison of 30 independent runs for benchmark functions f1–f15 ($D = 100$).

Functions	Run time of algorithms						
($D = 100$)	PSO	SA_GA	GSA	SCA	MVO	SOA	ECSOA
f_1	19.2614	542.3931	150.8607	**15.6790**	37.1642	45.5006	63.004
f_2	22.358433	521.8564	154.253122	**16.774635**	22.505080	54.167535	68.65450
f_3	**131.065635**	3509.5453	276.851090	157.841395	150.595822	423.272832	720.86686
f_4	21.427574	642.475529	154.476221	**17.295688**	40.792862	47.984464	67.91751
f_5	19.859456	651.808449	188.478202	**18.794625**	41.892321	54.515291	79.987520
f_6	20.542143	480.833828	153.064086	**17.318195**	39.282568	47.876927	68.786422
f_7	34.792130	890.981756	168.550287	**30.316487**	51.654684	90.520669	162.014
f_8	23.367437	481.638596	153.526963	**17.055855**	39.908609	55.511679	68.06153
f_9	35.632177	975.1530	173.578067	**32.396346**	45.258449	102.887695	120.19844
f_{10}	26.354619	562.020557	158.499353	**20.396761**	22.749008	60.474337	82.973951
f_{11}	**21.033824**	648.6857	184.680588	22.147268	41.646913	59.599374	70.32757
f_{12}	23.395074	624.441362	173.135274	**21.299206**	43.723453	64.893256	74.985
f_{13}	29.505900	615.5540	169.814804	**23.426107**	46.092607	63.673136	82.41825
f_{14}	60.288823	1654.958657	193.881593	**52.751386**	74.326149	165.947519	211.787815
f_{15}	59.640283	1629.496415	196.205228	**53.109655**	75.622178	166.651257	207.405906
The total time	548.524908	14431.84265	2649.855578	516.602609	773.214903	1503.476571	2149.389274
Overall rank	2	7	6	**1**	3	4	5

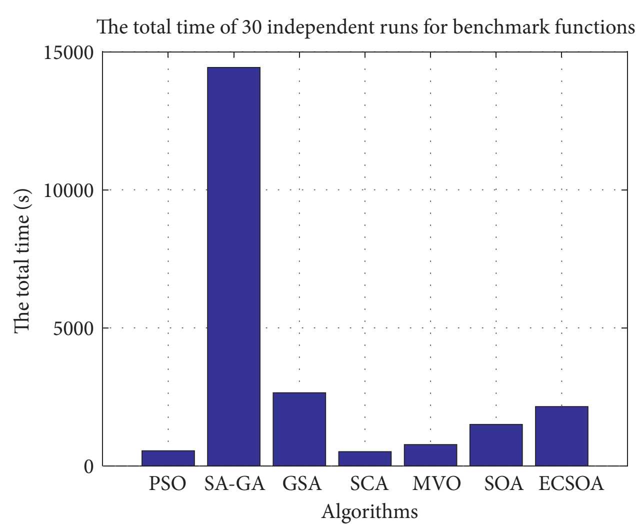

Figure 4: The total time of 30 independent runs of 7 algorithms on fifteen benchmark functions.

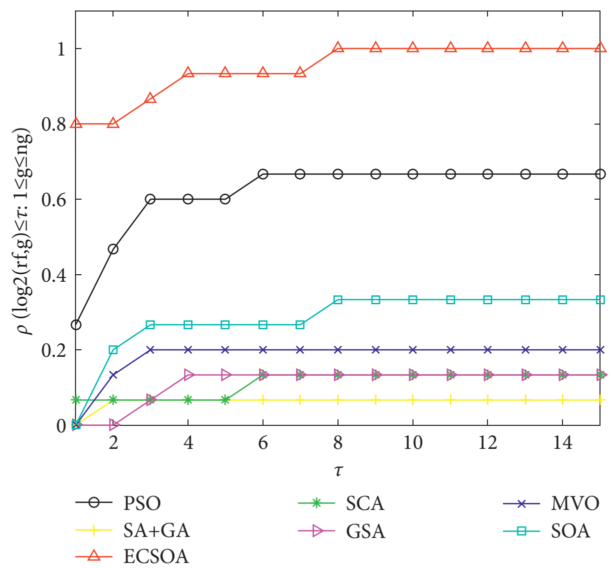

Figure 5: Performance profile of 7 algorithms on fifteen benchmark functions.

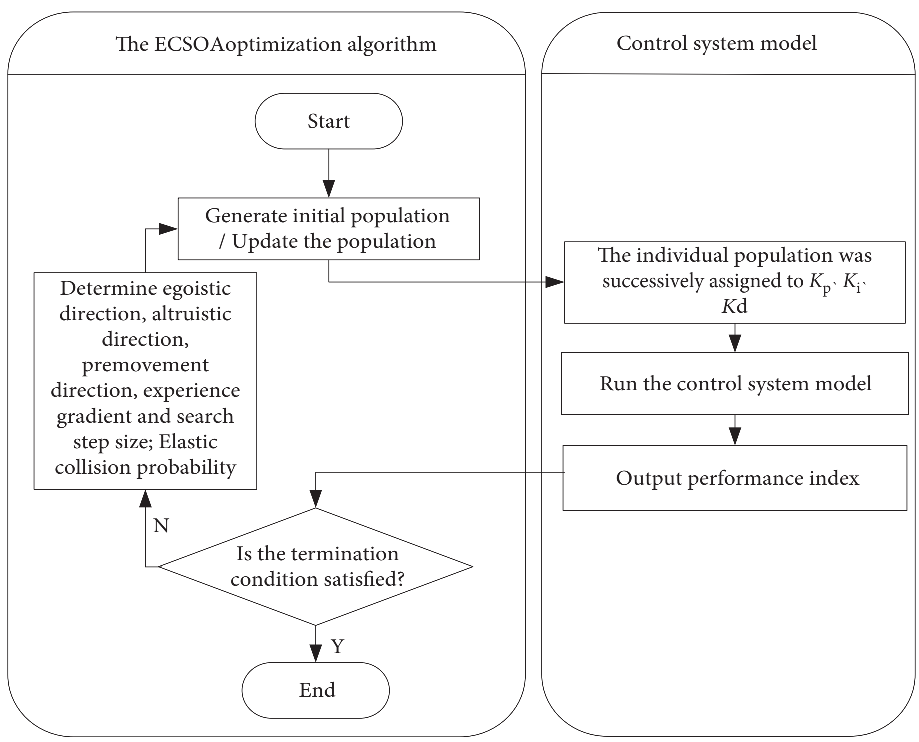

Figure 6: A process diagram for optimizing the test control system PID parameters by the ECSOA.

ECSOA, it is compared with the PSO, SA_GA, GSA, SCA, MVO, and SOA in terms of the PID controller parameter optimization. The mean values, standard deviation values, best fitness values, and best fitness values rank of the algorithms of 30 independent runs for g1–g4 are displayed in Table 8. The boldface indicates that the optimal result is better.

For the PID controller parameter optimization problems, according to Table 8, except g3 and g4, as to the best fitness, the ECSOA is better than the others. The optimal fitness value result of the ECSOA for g3 model is only worse than the SA_GA, the optimal fitness value result of the ECSOA for g4 model is only worse than the PSO algorithm. As to the standard deviation results, for g1 model, the ECSOA is only worse than the SA_GA, SCA, and the MVO; for g2 and g3 models, the ECSOA is only worse than the SA_GA; and for g4 model, the ECSOA is only worse than the MVO. Except for g1 and g4, as to the mean test results, the ECSOA is better than the others; for g1 model, the ECSOA is only worse than the SCA; and for g4 model, the ECSOA is only worse than the MVO. According to the optimal fitness value mean rank and all rank results from Table 8, the

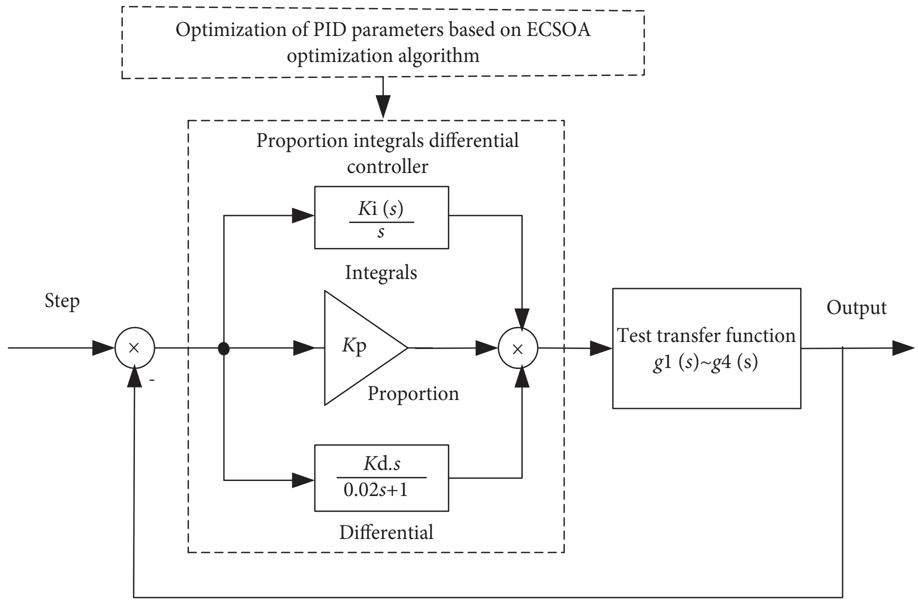

Figure 7: The optimization PID parameters model structure of the test control system.

Table 8: Performance comparison of algorithms in PID parameter optimization of 30 independent runs.

Test	Result	Algorithm						
		PSO	SA_GA	GSA	SCA	MVO	SOA	ECSOA
g_1	Mean	0.2267	0.3169	0.4571	**0.0918**	0.2501	0.1917	0.106057
	Std.	0.0877	0.0649	0.1569	**0.0263**	0.0532	0.11226	0.072471
	Best	0.0485	0.1002	0.2732	0.0483	0.0513	0.05774	**0.047885**
	Rank	3	6	7	2	4	5	**1**
g_2	Mean	58.4757	62.4599	60.7787	24.8454	59.5805	42.1538	**3.846738**
	Std.	7.75976	**0.1216**	5.3034	21.5239	7.6556	27.9025	2.706181
	Best	36.0409	62.0356	42.7711	0.4898	32.6095	0.39301	**0.310553**
	Rank	5	7	6	3	4	2	**1**
g_3	Mean	1.8481e+2	2.7179e+2	2.7665e+2	29.0458	1.0848e+2	2.6269e+2	**14.234925**
	Std.	59.6434	**0.62334**	10.3088	11.9839	56.6750	44.8106	2.840193
	Best	32.5445	**2.71191**	2.7139e+2	14.5588	20.0492	26.5763	8.737658
	Rank	6	**1**	7	3	4	5	2
g_4	Mean	1.7713e+2	55.3556	2.3413e+2	85.196656	**35.721213**	46.10528	35.837850
	Std.	4.2182e+2	36.00807	2.1754e+2	1.0050e+2	**1.411226**	26.992197	3.905932
	Best	**34.625063**	34.6294	58.321733	34.867448	34.643162	34.745734	34.625510
	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**

ECSOA can find solutions and has very strong robustness for the PID controller parameter optimization problems.

4.3.3. Convergence Curves Comparison of Algorithms in PID Controller Parameter Optimization. Figure 8 shows the fitness curves of PID controller parameter optimization for g1–g4. The comparison between the seven algorithms in Figure 8 shows that the convergence of the ECSOA is fast and the precision of the ECSOA is the best. The ECSOA can find the optimal value.

4.3.4. ANOVA Tests Comparison of Algorithms in PID Controller Parameter Optimization. Figure 9 is the ANOVA of the global best values PID controller parameter optimization for g1–g4. As seen from Figure 9, ECSOA is the most robust algorithm.

4.3.5. Unit Step Function PID Controller Parameter Optimization. Figure 10 shows the unit step function PID controller parameter optimization for g1–g4. As seen from Figure 10, the ECSOA is used to optimize the unit step function PID controller parameters of g1–g4, and the unit step functions tend to stabilize very quickly and accurately.

Therefore, the ECSOA is an effective and feasible solution in the control system models optimizing PID parameters.

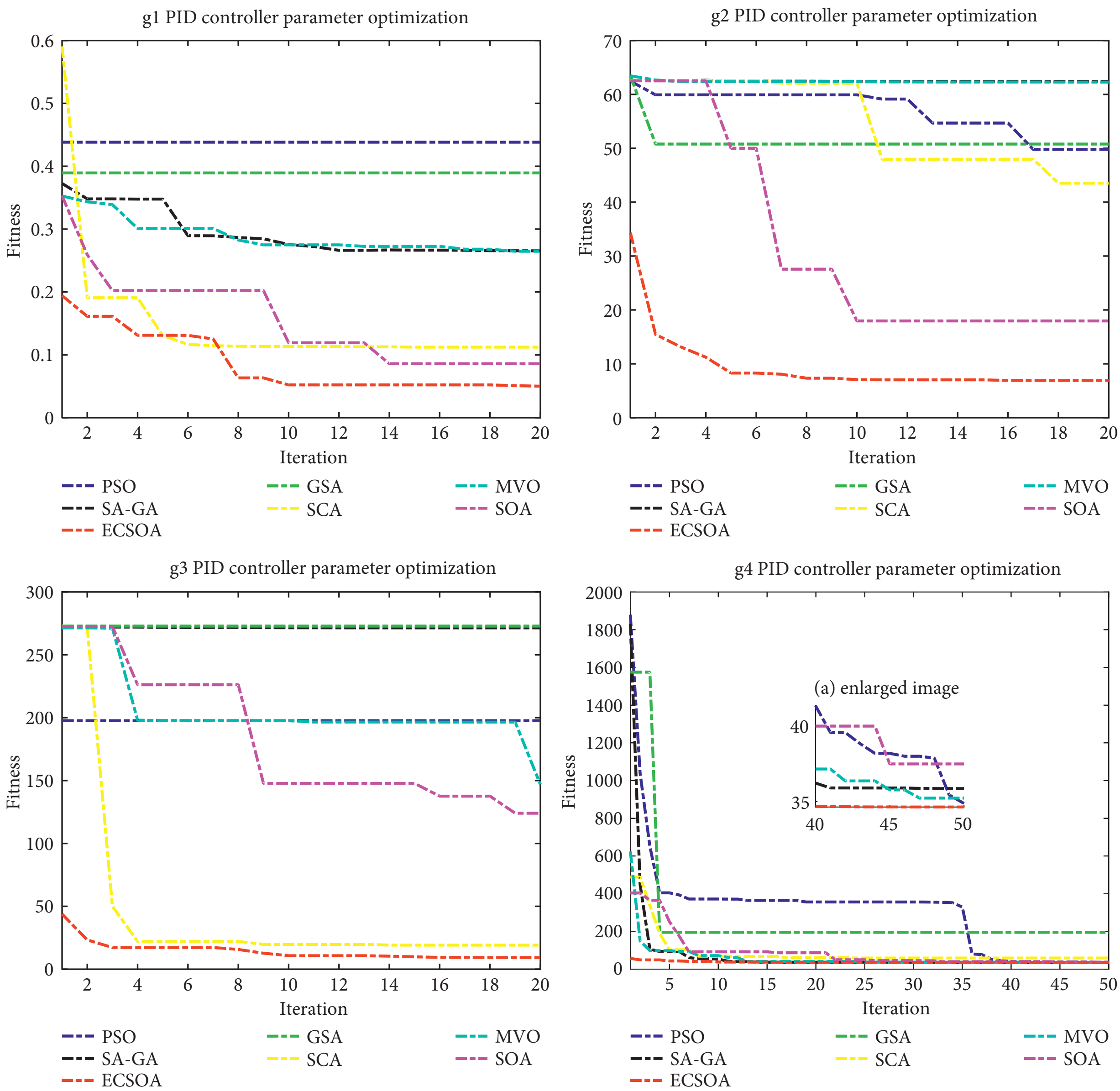

FIGURE 8: Convergence curves for PID controller parameter optimization, g1–g4.

4.4. Algorithm Performance Comparison in Constrained Engineering Optimization Problems. We are using six constrained engineering problems to test the capability of the ECSOA further. These constrained engineering problems are very popular in the literature. The penalty function is used to calculate the constrained problem. The parameters set for all of the heuristic algorithms still adopt the parameter setting in Table 4 of section 4.2.3. The formulations of these problems are available in the appendix.

4.4.1. Welded Beam Design Problem. This is a least fabrication cost problem, which has four parameters and seven constraints. The parameters of the structural system are shown in Figure 11 [7]. Some of the algorithms are taken from other literature as follows: GSA [6], MFO [7], MVO [9], CPSO [45], and HS [46]. For the problem in this paper, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best-obtained values in Table 9.

In Table 9, the ECSOA is better than GSA, MFO, MVO, GA, CPSO, and HS algorithms in other literature. The ECSOA is also better than the PSO, SA_GA, GSA, SCA, MVO, and SOA. Therefore, the ECSOA can resolve the problem.

4.4.2. Pressure Vessel Design Problem. This is also the least fabrication cost problem of four parameters and four constraints. The parameters of the structural system are shown in Figure 12 [7]. Some of the algorithms are taken from other literature as follows: MFO [7], ES [47], DE [48], ACO [49], and GA [50]. For the problem, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best-obtained values in Table 10.

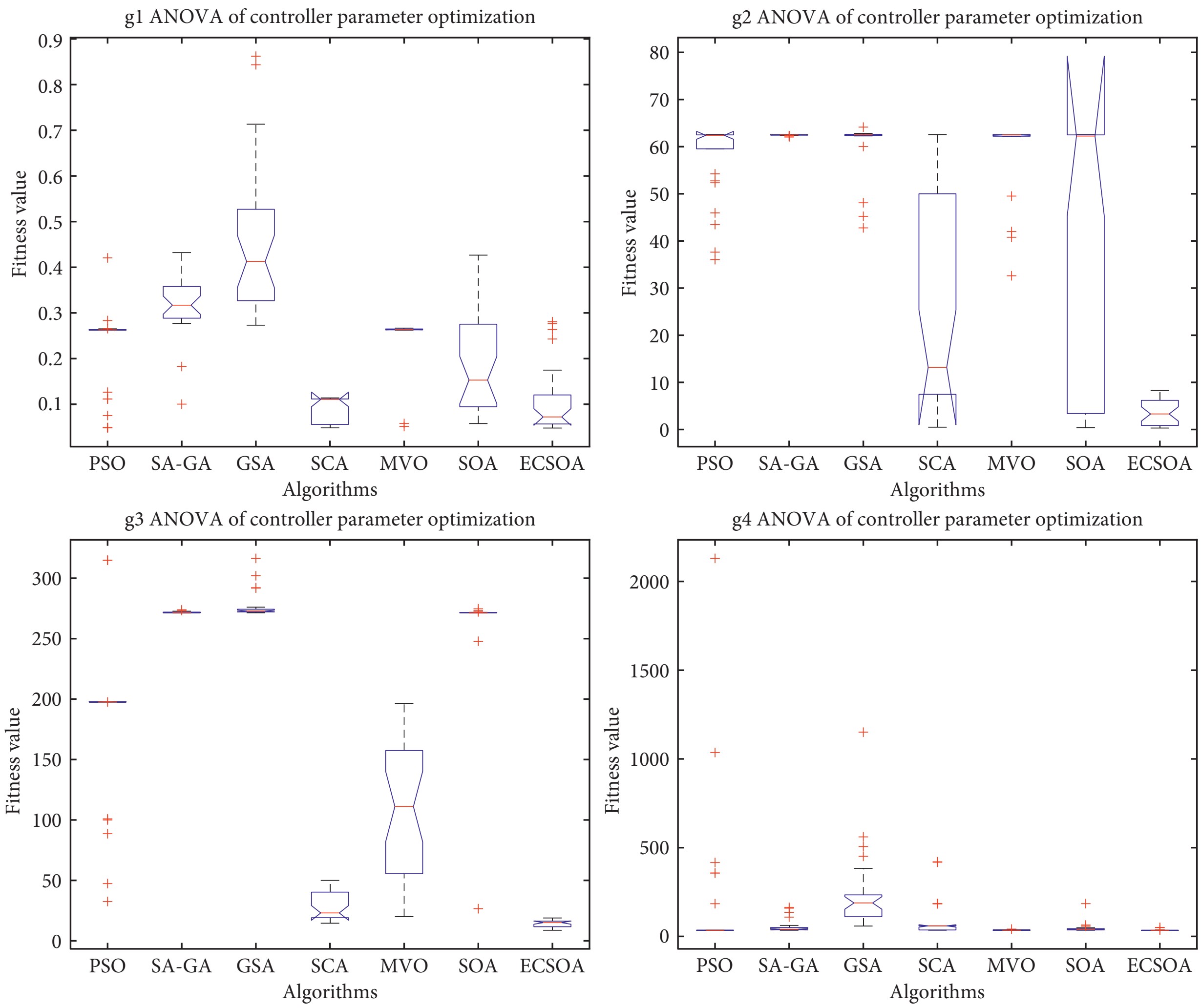

Figure 9: The ANOVA tests for PID controller parameter optimization, g1–g4.

For the problem, the ECSOA is better than the MFO, ES, DE, ACO, and GA algorithms in other literature. The ECSOA is also better than the PSO, SA_GA, GSA, SCA, and MVO. There is not much difference between the optimal value of ESOA and that of SOA. Therefore, ECSOA can resolve the problem.

4.4.3. Cantilever Beam Design Problem. This is a problem that is determined by five parameters and is only applied to the scope of variables of constraints. The parameters of the structural system are shown in Figure 13 [7]. Some of the algorithms are taken from other literature as follows: MFO [7], CS [51], GCA [52], MMA [52], and SOS [53]. For the problem, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best-obtained values in Table 11.

In Table 11, the ECSOA proves to be better than the MFO, CS, GCA, MMA, and SOS algorithm in other literature. The ECSOA is also better than the PSO, SA_GA, GSA, SCA, and MVO. There is not much difference between the optimal value of ECSOA and that of SOA. Therefore, the ECSOA can resolve the problem.

4.4.4. Gear Train Design Problem. This is a minimum gear ratio problem, which has four variables and a scope of variables of constraints. Figure 14 is the schematic diagram [7]. Some of the algorithms are taken from other literature as follows: MFO [7], MVO [9], CS [51], ABC [54], and MBA [54]. For the problem in this paper, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best-obtained values in Table 12.

In Table 12, the ECSOA proves to be better than the MFO, MVO, CS, ABC, and MBA algorithm in other literature. Except for the SA_GA, GSA, and PSO, the ECSOA is also better than the SCA, the MVO, and the SOA. The result of the ECSOA has reached the theoretical best solution, although the optimum of the ECSOA is worse than that of the SA_GA, GSA, and PSO. The ECSOA finds a new value. Therefore, the ECSOA can resolve the problem.

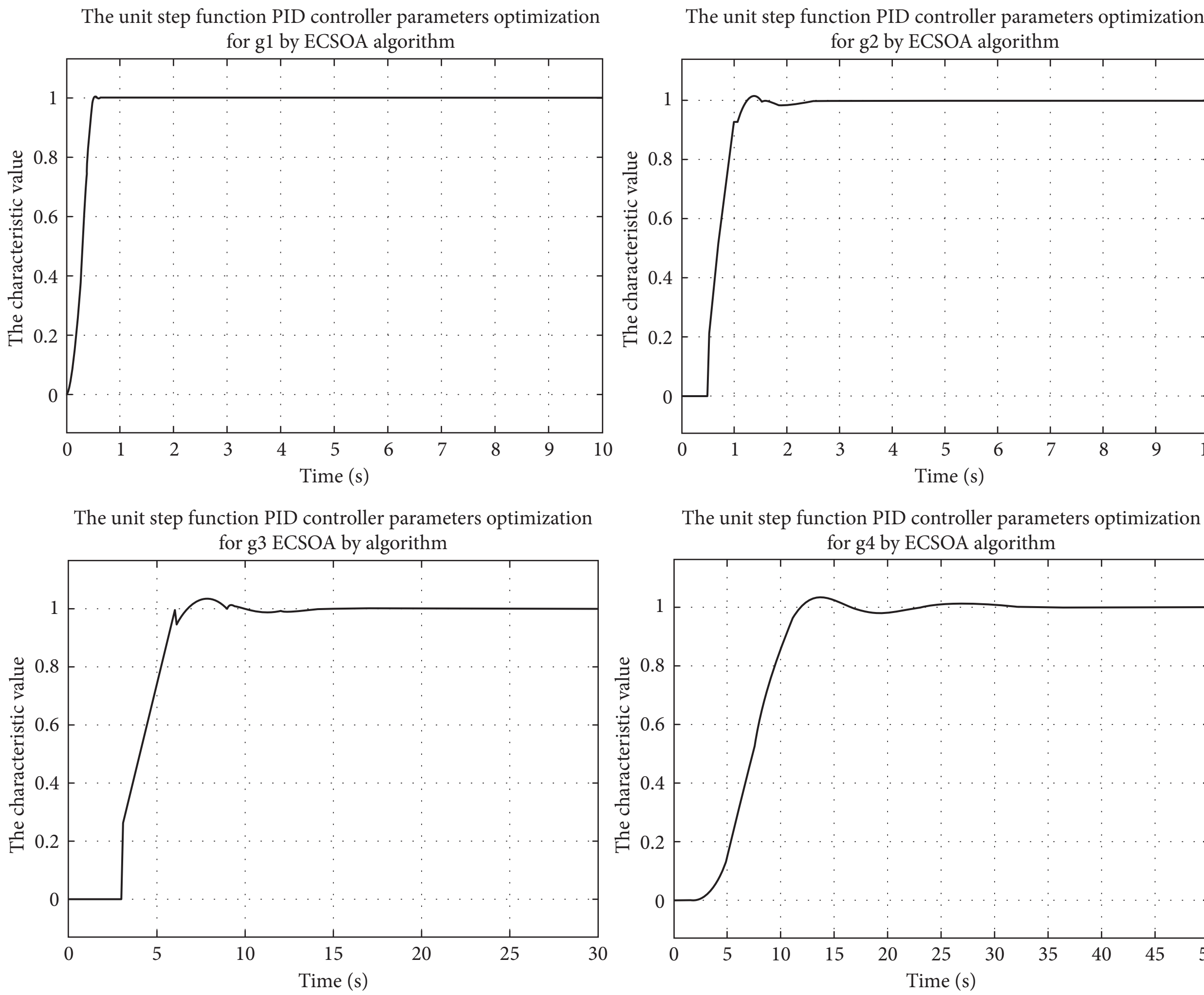

FIGURE 10: The unit step function PID controller parameter optimization, g1–g4.

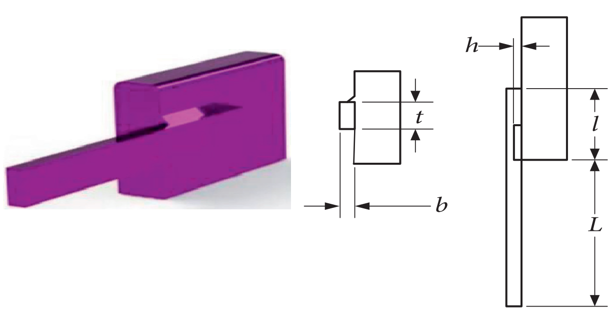

FIGURE 11: Design parameters of the welded beam design problem.

4.4.5. Three-Bar Truss Design Problem. This is a minimize weight problem under stress, which has two variables and only applies to the scope of the variables of constraints. Figure 15 is the schematic diagram of the components [7]. Some of the algorithms are taken from other literature as follows: MFO [7], MVO [9], CS [51], MBA [54], and DEDS [55]. For the problem, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best values in Table 13.

In Table 13, except for the MVO and the PSO, the ECSOA is better than the others. The best value of the ECSOA has reached the theoretical best solution, although the optimum of the ECSOA is worse than that of the MVO and the PSO. Therefore, the ECSOA can resolve the problem.

4.4.6. I-Beam Design Problem. This is a minimize vertical deflection problem that has four variables and a constraint. Figure 16 is the design diagram [7]. Some of the algorithms are taken from other literature as follows: MFO [7], CS [51], SOS [53], IARSM [56], and ARSM [56]. For the problem, the ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA and provides the best-obtained values in Table 14.

In Table 14, except for the MFO, GSA, SOA, and SA_GA, the ECSOA is better than the others. The fitness of the MFO is the best. Although the most minor vertical deviation of the ECSOA is not as good as that of the GSA, the SOA, and the SA_GA, it is very close to other relative optimal values. Therefore, the ECSOA is an effective and feasible solution to the I-beam design optimization problem.

In brief, the ECSOA proves to be better than the other algorithms in most actual studies. The ECSOA can resolve these practical problems.

Table 9: Comparison results of the welded beam design problem.

Algorithm	Optimal values for variables h	l	t	b	Optimal cost	Rank
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 [45]	0.202369	3.544214	9.048210	0.205723	1.72802	7
HS [46]	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
ECSOA	0.18588842973	3.68013819994	9.06091584266	0.20569607018	**1.69693487297**	**1**

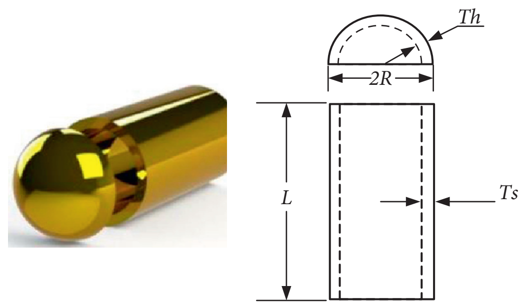

Figure 12: Pressure vessel design problem.

Table 10: Comparison results for pressure vessel design problem.

Algorithm	Optimal values for variables T_s	T_h	R	L	Optimal cost	Rank
MFO [9]	0.8125	0.4375	42.098445	176.636596	6059.7143	8
ES [47]	0.8125	0.4375	42.098087	176.640518	6059.7456	10
DE [48]	0.8125	0.4375	42.098411	176.637690	6059.7340	9
ACO [49]	0.8125	0.4375	42.103624	176.572656	6059.0888	7
GA [50]	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**	**1**
ECSOA	1.08847662958	0.52082333958	57.69587709181	47.007639941708	5736.5315190781	2

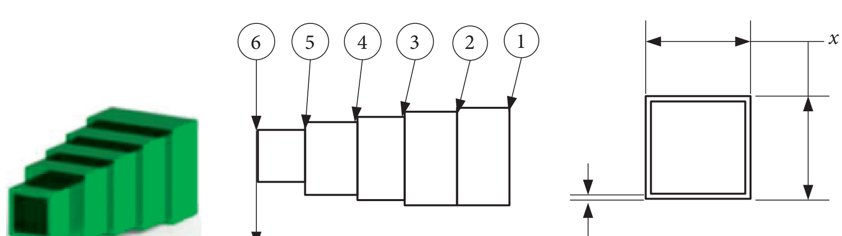

Figure 13: Cantilever beam design problem.

Table 11: 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 [51]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999	7
GCA [52]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	8
MMA [52]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400	8
SOS [53]	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**	**1**
ECSOA	5.993351697	5.332166494	4.470567257	3.5421087592	2.137286902	1.339957510	2

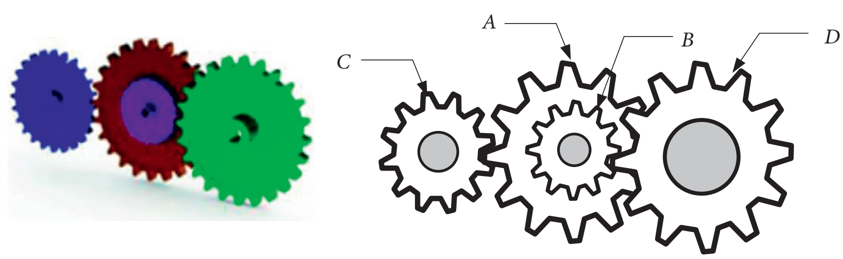

Figure 14: Gear train design problem.

Table 12: 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 [51]	43	16	19	49	2.7009e-012	7
ABC [54]	49	16	19	43	2.7009e-012	7
MBA [54]	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
ECSOA	58.3864369781	18.85988647288	21.2811710819	47.6451311622	8.240738329e-19	4

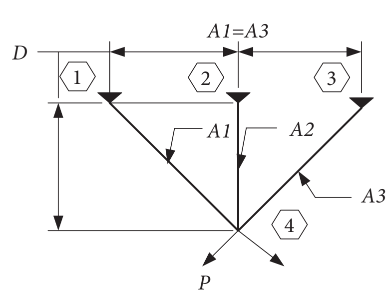

Figure 15: Three-bar truss design problem.

Table 13: Comparison results of the three-bar truss design problem.

Algorithm	Optimal values for variables x_1	x_2	Optimum weight	Rank
MFO [7]	0.788244770931922	0.409466905784741	263.895979682	10
MVO [9]	0.78860276	0.40845307	263.8958499	8
CS [51]	0.78867	0.40902	263.9716	11
MBA [54]	0.7885650	0.4085597	263.8958522	9
DEDS [55]	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
ECSOA	0.788380881070123	0.408297927805087	263.8523494157003	3

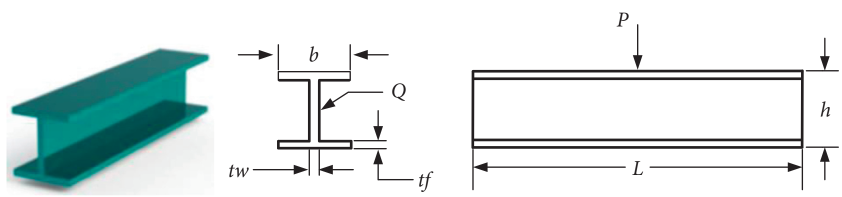

Figure 16: I-beam design problem.

Table 14: Comparison results for I-beam design problem.

Algorithm	Optimal values for variables b	h	t_w	t_f	Optimum vertical deflection	Rank
MFO [7]	50	80	1.7647	5.0000	**0.0066259**	**1**
CS [51]	50	80	0.9 5	2.32167	0.0130747	9
SOS [53]	50	80	0.9	2.32179	0.0130741	8
IARSM [56]	48.42	79.99	0.90	2.40	0.131	11
ARSM [56]	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	5
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
ECSOA	34.9997548457	79.9999999921	4.9999998123	4.9999999961	0.0078636983	3

5. Conclusion

An ECSOA is presented, with a completely elastic collision, completely inelastic collision, and non-completely elastic collision method. According to the four-phase test of the ECSOA from different perspectives, it improved the SOA, the benchmark function optimization, the PID control parameter optimization problems, and the constrained engineering problems.

In the first phase, the SOA is improved in seven different ways: the SOA based on parameter change (PCSOA), the SOA based on parameter adaptive Gaussian transform (PAGTSOA), the SOA based on Levy variation (LVSOA), the SOA based on refraction reverse learning mechanism (RRLOOA), the SOA based on mutual benefit factor strategy (MBFSOA), the SOA based on Cauchy variation (CVSOA), and the SOA based on elastic collision (ECSOA). Each improved algorithm was optimized for the fifteen functions. The result is that the ECSOA is feasible in the benchmark functions. In this phase, we consider the ranking values of 30 independent runs between the ECSOA mean values, the standard deviation values, the best fitness values, the best fitness values rank, the convergence curves, and the variance tests for the global minimum values.

In the second phase, fifteen benchmark function optimization problems are used to test the ECSOA further. The ECSOA is compared to the PSO, SA_GA, GSA, SCA, MVO, and SOA for verification. It was observed that the ECSOA is feasible and competitive in benchmark functions. The second test phase is also about the ranking values of 30 independent runs between the ECSOA mean values, standard deviation values, best fitness values, best fitness values rank, convergence curves, and variance tests for the global minimum values. In the benchmark function optimization problems, the complexity analysis of the ECSOA is researched, and the overall calculational complexity of the ECSOA is almost the same as that of the basic SOA. Wilcoxon's rank-sum test is studied, and the ECSOA proves to be better than the other six algorithms. Based on the run time comparison of seven algorithms in benchmark functions, the ECSOA has relatively more program running time, and it is not optimal in terms of running time. From the results of the performance ratios of the average solution for the seven algorithms, the optimization probability of the ECSOA is the highest.

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

Eventually, in the last phase, six engineering problems further tested the ECSOA. The ECSOA was compared to various algorithms. The results prove that the ECSOA is the highest competitive algorithm for the practical optimization problems.

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

(1) The elastic collision strategy includes the completely elastic collision, the completely inelastic collision, and the noncomplete elastic collision. The three different situations of elastic collision strategy tend to generate random seekers, increase the diversity of the seeker, increase the search space, and avoid premature convergence.

(2) Among the eight improved algorithms (PCSOA, PAGTSOA, ECSAO, LVSOA, RRLOOA, MBFSOA, CVSOA, and ECSOA), the ECSOA performed best in the benchmark functions test.

(3) Among the seven algorithms (PSO, SA_GA, GSA, SCA, MVO, SOA, and ECSOA), the ECSOA optimization benchmark function has the highest optimization capability.

(4) The ECSOA optimization benchmark functions have almost the same calculational complexity as the SOA.

(5) The running time of the ECSOA optimization benchmark function is relatively high. Among the seven algorithms compared, the running time is only better than that of the SA_GA.

(6) The ECSOA can solve real challenging problems, such as the PID control parameter optimization problems and the classical constrained engineering optimization problems.

(7) Further improving and application can be incorporated into future studies. The improved SOA and the heuristic algorithms based on those improved strategies can be applied not only to engineering optimization problems, but also to path planning problems, pattern recognition, intelligent control and other fields, and many practical application optimization problems that cannot be solved by traditional methods. Except the methods used in the paper, some of representative computational intelligence algorithms can be used to solve the problems, such as the MBO, EHO, MS, SMA, and HHO.

Appendix

A. Welded Beam Design Problem

Consider $\overrightarrow{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$, and minimize $f(\overrightarrow{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$, subject to

$$g_1(\overrightarrow{x}) = \tau(\overrightarrow{x}) - \tau_{\max} \leq 0, \tag{A.1}$$

$$g_2(\overrightarrow{x}) = \sigma(\overrightarrow{x}) - \sigma_{\max} \leq 0, \tag{A.2}$$

$$g_3(\overrightarrow{x}) = x_1 - x_4 \leq 0, \tag{A.3}$$

$$g_4(\overrightarrow{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0, \tag{A.4}$$

$$g_5(\overrightarrow{x}) = 0.125 - x_1 \leq 0, \tag{A.5}$$

$$g_6(\overrightarrow{x}) = \delta(\overrightarrow{x}) - \delta_{\max} \leq 0, \tag{A.6}$$

$$g_7(\overrightarrow{x}) = \mathrm{P} - P_c(\overrightarrow{x}) \leq 0. \tag{A.7}$$

Variable ranges are $0.1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$, $0.1 \leq x_3 \leq 10$, and $0.1 \leq x_4 \leq 2$, where

$$\tau(\overrightarrow{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''(x_2/2R) + (\tau'')^2}$$
$$\tau' = (P/\sqrt{2}\,x_1x_2),\ \tau'' = (MP/J),\ M = \mathrm{P}(\mathrm{L} + (x_2/2))$$
$$R = \sqrt{(x_2^2/4) + (x_1 + x_3/2)^2}$$
$$J = 2\left\{\sqrt{2}\,x_1x_2\left[(x_2^2/4) + (x_1 + x_3/2)^2\right]\right\}$$
$$\sigma(\overrightarrow{x}) = (6\mathrm{PL}/x_4x_3^2)$$
$$\delta(\overrightarrow{x}) = (6\mathrm{PL}^3/Ex_4x_3^3)$$

$$P_c(\overrightarrow{x}) = (4.013\mathrm{E}\sqrt{x_3^2x_4^6/36}\,/\mathrm{L}^2)(1 - (x_3/2\mathrm{L})\sqrt{(\mathrm{E}/4\mathrm{G})})$$

$P = 6000$ lb, $L = 14$ in, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 136000$ psi, $\sigma_{\max} = 30000$ psi, $\delta_{\max} = 0.25$ in

B. Pressure Vessel Design Problem

Consider $\overrightarrow{x} = [x_1, x_2, x_3] = [T_s, T_h, R, L]$, and minimize $f(\overrightarrow{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$, subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0, \tag{B.1}$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0, \tag{B.2}$$

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

$$g_4(\vec{x}) = x_4 - 240 \le 0. \tag{B.4}$$

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

C. Cantilever Design Problem

Consider $\vec{x} = [x_1, x_2, x_3, x_4, x_5]$, and minimize $f(\vec{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$, subject to $g(\vec{x}) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 - 1 \le 0$.

Variable ranges are $0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100$.

D. Gear Train Design Problem

Consider $\vec{x} = [x_1, x_2, x_3, x_4] = [n_A, n_B, n_C, n_D]$, and minimize $f(\vec{x}) = (1/6.931 - x_3x_2/x_1x_4)^2$.

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

E. Three-Bar Truss Design Problem

Consider $\vec{x} = [x_1, x_2] = [A_1, A_2]$, and minimize $f(\vec{x}) = (2\sqrt{2}\,x_1 + x_2) * \mathrm{l}$, subject to

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}\mathrm{P} - \sigma \le 0, \tag{E.1}$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}\mathrm{P} - \sigma \le 0, \tag{E.2}$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1}\mathrm{P} - \sigma \le 0. \tag{E.3}$$

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

F. I-Beam Design Problem

Consider $\vec{x} = [x_1, x_2, x_3, x_4] = [b, h, t_w, t_f]$, and minimize $f(\vec{x}) = 5000/x_3(x_2 - 2x_4)\quad{}^3/12 + x_1x_4^3/6 + 2x_1x_4(x_2 - x_4/2)^2$, subject to $g(\vec{x}) = 2x_1x_3 - x_3(x_2 - 2x_4) \le 0$.

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

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 51766005 and 52166001) and the Science and Technology Project of Yunnan Tobacco Company of China (Grant no. 2019530000241019).

References

[1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
[2] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, 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] E. H. L. Aarts and P. J. M. Laarhoven, "Simulated annealing: an introduction," *Statistica Neerlandica*, vol. 43, no. 1, pp. 31–52, 1989.
[5] Z. W. Zong Woo Geem, J. H. Joong Hoon Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
[6] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "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*, vol. 89, pp. 228–249, 2015.
[8] S. Mirjalilia, "A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 27, pp. 1–14, 2016.
[9] S. Mirjalilia, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing & Applications*, vol. 17, pp. 16–19, 2015.
[10] M. Tuba, I. Brajevic, and R. Jovanovic, "Hybrid seeker optimization algorithm for global optimization," *Applied Mathematics & Information Sciences*, vol. 7, no. 3, pp. 867–875, 2013.
[11] J. Ji, S. Song, C. Tang, S. Gao, Z. Tang, and Y. Todo, "An artificial bee colony algorithm search guided by scale-free networks," *Information Sciences*, vol. 473, pp. 142–165, 2019.
[12] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
[13] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.
[14] G.-Ge Wang, S. Deb, and L. dos S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence*, pp. 1–5, Bali, Indonesia, December 2015.
[15] G.-Ge Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 9, pp. 1–14, 2016.
[16] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
[17] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and

applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
[18] B. S. Yildiz and A. R. Yildiz, "The Harris hawks optimization algorithm, salp swarm algorithm, grasshopper optimization algorithm and dragonfly algorithm for structural design optimization of vehicle components," *Materials Testing*, vol. 61, pp. 744–748, 2019.
[19] H. Jia, Z. Wei, and X. He, "Path planning based on improved particle swarm optimization algorithm," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 49, no. 12, pp. 371–377, 2018.
[20] L. d. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.
[21] X. Song, L. Jian, and Y. Song, "A chunk updating LS-SVMs based on block Gaussian elimination method," *Applied Soft Computing*, vol. 51, pp. 96–104, 2017.
[22] Y. Saji and M. Barkatou, "A discrete bat algorithm based on Levy flights for Euclidean traveling salesman problem," *Expert Systems with Applications*, vol. 172, pp. 1–17, 2021.
[23] E. Shadkam, "Cuckoo optimization algorithm in reverse logistics: a network design for COVID-19 waste management," *Waste Management & Research: The Journal for a Sustainable Circular Economy*, pp. 1–12, 2021.
[24] M. kelidari and J. Hamidzadeh, "Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator," *Soft Computing*, vol. 25, no. 4, pp. 1–23, 2021.
[25] Y. Wang and Z. Ma, "Elite symbiotic organisms search algorithm based on subpopulation stretching operation," *Control and Decision*, vol. 34, no. 7, pp. 1355–1364, 2019.
[26] X. Chu, S. Li, D. Gao, W. Zhao, J. Cui, and L. Huang, "A binary superior tracking artificial bee colony with dynamic Cauchy mutation for feature selection," *Complexity*, vol. 2020, pp. 6–25, 2020.
[27] Z. Liu, S. Li, B. Li, and H. Wang, "New elastic collision optimization algorithm and its application in sensor cloud resource scheduling," *Journal of Zhejiang University (Engineering Science)*, vol. 52, no. 8, pp. 1431–1443, 2018.
[28] C. Dai, Y. Zhu, and W. Chen, "Seeker optimization algorithm," in *Proceedings of the.2006 International Conference Computational Intelligence and Security*, pp. 225–229, IEEE Press, Guangzhou, China, November 2006.
[29] C. Dai, Y. Zhu, and W. Chen, "Seeker optimization algorithm," *Computational Intelligence and Security*, vol. 4456, pp. 167–176, 2007.
[30] C. Chaohua Dai, W. Weirong Chen, Y. Yunfang Zhu, and X. Xuexia Zhang, "Seeker optimization algorithm for optimal reactive power dispatch," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1218–1231, 2009.
[31] C. Dai, W. Chen, Y. Song, and 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.
[32] C. Chaohua Dai, W. Weirong Chen, and Y. Yunfang Zhu, "Seeker optimization algorithm for digital IIR filter design," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1710–1718, 2010.
[33] C. Dai, W. Chen, Y. Zhu, Z. Jiang, and Z. You, "Seeker optimization algorithm for tuning the structure and parameters of neural networks," *Neurocomputing*, vol. 74, no. 6, pp. 876–883, 2011.
[34] C. Dai, W. Chen, Z. Cheng, Q. Li, Z. Jiang, and 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 & Energy Systems*, vol. 33, no. 3, pp. 369–376, 2011.
[35] C. Dai, W. Chen, L. Ran, Y. Zhang, and Y. Du, "Human group optimizer with local search," *Lecture Notes in Computer Science*, vol. 6728, pp. 310–320, 2011.
[36] Y. Zhu, C. Dai, and W. Chen, "Seeker optimization algorithm for several practical applications," *International Journal of Computational Intelligence Systems*, vol. 7, no. No. 2, pp. 353–359, 2014.
[37] 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.
[38] X. Li, J. Zhang, and M. Yin, "Animal migration optimization: an optimization algorithm inspired by animal migration behavior," *Neural Computing & Applications*, vol. 24, no. 7-8, pp. 1867–1877, 2014.
[39] M. Molga and C. Smutnicki, "Test functions for optimization needs," 2005, http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf.
[40] J. Kennedy, *Particle Swarm Optimization in Encyclopedia of Machine Learning*, pp. 760–766, Springer, New York, NY, USA, 2010.
[41] H. Yu, H. Fang, P. Yao, and Y. Yuan, "A combined genetic algorithm/simulated annealing algorithm for large scale system energy integration," *Computers & Chemical Engineering*, vol. 24, no. 8, pp. 2023–2035, 2000.
[42] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*, pp. 15–18, China Machine Press, New York, NYUnited States, 2009.
[43] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
[44] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
[45] R. A. Krohling and L. dos Santos Coelho, "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 6, pp. 1407–1416, 2006.
[46] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36-38pp. 3902–3933, 2005.
[47] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *International Journal of General Systems*, vol. 37, no. 4, pp. 443–473, 2008.
[48] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," *Computers & Structures*, vol. 85, no. 7-8, pp. 340–349, 2007.
[49] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Engineering Computations*, vol. 27, no. 1, pp. 155–182, 2010.
[50] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 11, pp. 2013–2015, 1991.
[51] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural

optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[52] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 5, pp. 829–846, 1996.

[53] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.

[54] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2592–2612, 2013.

[55] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.

[56] G. G. Wang, "Adaptive response surface method using inherited Latin hypercube design points," *Journal of Mechanical Design*, vol. 125, no. 2, pp. 210–220, 2003.