Hindawi
Complexity
Volume 2020, Article ID 4968063, 24 pages
https://doi.org/10.1155/2020/4968063

Research Article

An Enhanced Comprehensive Learning Particle Swarm Optimizer with the Elite-Based Dominance Scheme

Chengcheng Chen,[1,2] Xianchang Wang,[1,2,3] Helong Yu,[4] Nannan Zhao,[4] Mingjing Wang,[5] and Huiling Chen[6]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China*
[3]*Chengdu Kestrel Artificial Intelligence Institute, Chengdu 611730, China*
[4]*College of Information Technology, Jilin Agricultural University, Changchun 130118, China*
[5]*Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[6]*College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, Zhejiang, China*

Correspondence should be addressed to Xianchang Wang; xcwang89@jlu.edu.cn, Helong Yu; yuhelong@aliyun.com, and Huiling Chen; chenhuiling.jlu@gmail.com

Received 25 July 2020; Accepted 17 September 2020; Published 20 October 2020

Academic Editor: Narayanan Kumarappan

In recent years, swarm-based stochastic optimizers have achieved remarkable results in tackling real-life problems in engineering and data science. When it comes to the particle swarm optimization (PSO), the comprehensive learning PSO (CLPSO) is a well-established evolutionary algorithm that introduces a comprehensive learning strategy (CLS), which effectively boosts the efficacy of the PSO. However, when the single modal function is processed, the convergence speed of the algorithm is too slow to converge quickly to the optimum during optimization. In this paper, the elite-based dominance scheme of another well-established method, grey wolf optimizer (GWO), is introduced into the CLPSO, and the grey wolf local enhanced comprehensive learning PSO algorithm (GCLPSO) is proposed. Thanks to the exploitative trends of the GWO, the algorithm improves the local search capacity of the CLPSO. The new variant is compared with 15 representative and advanced algorithms on IEEE CEC2017 benchmarks. Experimental outcomes have shown that the improved algorithm outperforms other comparison competitors when coping with four different kinds of functions. Moreover, the algorithm is favorably utilized in feature selection and three constrained engineering construction problems. Simulations have shown that the GCLPSO is capable of effectively dealing with constrained problems and solves the problems encountered in actual production.

1. Introduction

Optimization problems are common problems in real life, and we need to achieve the best solution when tackling a specific problem. With the increase of complexity of the problem, the traditional gradient-based method is difficult to better optimize some types of problems [1, 2]. To deal with this problem, metaheuristic algorithms are widely used in real life. These algorithms use iterations and randomly generate optimal solutions for optimization problems by simulating natural phenomena or social behaviors [3–6]. The underlying idea behind these technologies is using mathematical algorithms to simulate biological and physical systems in nature, such as natural evolutionary and swarm intelligence algorithms. Previous studies confirm that metaheuristic algorithms possess more effectiveness than gradient-based algorithms in coping with some problems that involve optimization [7–10]. Also, the metaheuristic algorithm (MA) has some weaknesses to be improved. For example, convergence to an optimal solution is relatively slow, and there is no universal model when dealing with different problems. Therefore, it is required to modify and enhance the core exploratory and exploitative abilities of stochastic algorithms on some optimization problems [11–18].

MA can be divided into swarm intelligence algorithms and evolutionary algorithms (EA). In detail, the inspirations of EAs come from the biological evolution process, and the mechanisms of competition and elimination are added into the algorithms. The representative evolutionary algorithms include differential evolution (DE) [19], genetic algorithm (GA) [20], and evolutionary strategy (ES). The group behavior of animals inspires swarm intelligence algorithms, for example, ant colony optimization (ACO) [21, 22], firefly algorithm (FA) [23], whale optimization algorithm (WOA) [24–28], moth-flame optimization (MFO) [29–31], grasshopper optimization algorithm (GOA) [32–34], bat algorithm (BA) [35, 36], moth search algorithm (MSA) [37], Harris hawks optimization [38], slime mould algorithm [39], and so on [40–44].

PSO [45] is a MA proposed by Eberhart et al. in 1995, which is motivated by the communication behaviors and social interaction of animals. PSO simulates the hunting behavior of the birds that cooperatively search for food. Each member of the group adjusts their search model by learning their own experience or other members. Inspired by this phenomenon, a mathematical model was established. In the PSO algorithm, a particle means a number of the group, which is a potential solution of the problem optimized and represents a point in the search space. The position of food is regarded as globally optimal. Each particle possesses a fitness value and speed, which can be adjusted according to the global optimal solution and the individual optimal solution. Due to its small number of parameters and ease of use, the PSO algorithm was used in function optimization [46], filter design [47], proportional-integral-derivative (PID) control [48], power allocation [49], and other scientific and engineering applications [50–53]. However, the algorithm tends to be trapped in local optimization when encountering complex multimodal problems. With the purpose of improving the performance of the algorithm, researchers have come up with a large number of PSO variants to promote the acceleration coefficient and inertia weight of the parameters controlled [54, 55] and applied the population number [56, 57] to the optimization problem. Gong et al. [58] added GA to PSO to promote the convergence performance of the algorithm. Zhan et al. [59] added the orthogonal learning (OL) strategy to PSO to promote the capacity of the algorithm to escape from the local optimum. Cheng and Jin [60] combined competitive learning strategies with PSO to better the convergence accuracy of the algorithm. To enhance the ability of the PSO algorithm to obtain the optimum on complex multimodal problems, Liang et al. [61] proposed a new, improved PSO algorithm, which is CLPSO. It adopted an innovative comprehensive learning strategy (CLS), where the individual best position of all particles is used to update the particle's speed. This mechanism allows group diversity to be conserved to prevent convergence in the premature period. However, the convergence speed of CLPSO on the unimodal function is very slow. For the sake of making the algorithm to converge to the optimal solution, it is necessary to enhance the local search ability of the algorithm near the optimum. Grey wolf optimizer (GWO) [62] is a MA proposed for global optimization. Its inspiration comes from the hunting process of grey wolves in nature. GWO uses the same principle to organize different individuals in the algorithm after learning the wolf group organization hierarchy. Since the GWO has fewer parameters, on the contrary, this strategy is relatively simple, flexible, and scalable, and the algorithm has a good convergence effect on the unimodal function. At present, this method has been applied in many fields, including neural network [15, 16, 63, 64], environment [65], medical diagnosis [17, 66–68], and image processing [69]. In this research, the GWO algorithm is introduced into CLPSO to generate a novel algorithm called GCLPSO, which can reach a certain harmony between local search and global search to enhance the ability of the algorithm when finding the optimum. Specifically, CLPSO can effectively preserve the population diversity and evade premature convergence. Then, GWO is utilized to perform a local search for excellent particles in CLPSO to achieve high convergence speed and accuracy. Theoretically, the proposed mechanism can greatly better the balance between exploration and development so that the algorithm can quickly converge to the optimum.

To analyze the efficiency of the algorithm, the benchmarks in CEC2017 [70] were adopted to evaluate the performance of the GCLPSO and other comparison algorithms. The comparison algorithms include seven MAs, such as PSO, dragonfly algorithm (DA) [71], GOA, sine cosine algorithm (SCA) [72], MFO, WOA, and GWO, and eight advanced evolutionary algorithms, such as Cauchy and Gaussian sine cosine optimization (CGSCA) [73], sine cosine algorithm with differential evolution algorithm (SCADE) [74], chaotic fruit fly optimization algorithm (CIFOA) [75], adaptive mutation fruit fly optimization algorithm (AMFOA) [76], Lévy flight trajectory-based whale optimization algorithm (LWOA) [77], improved whale optimization algorithm (IWOA) [78], biogeography-based learning particle swarm optimization (BLPSO) [79], and CLPSO [61]. Experimental results have shown that the improved algorithm is considerably superior to other comparison algorithms in finding the optimal solution. Also, GCLPSO has shown a good effect on the engineering constraint problems. This paper applies the proposed algorithm to the problems of the pressure vessel, welded beam, and I-beam design models. It can be seen from the optimization outcomes of the comparisons that the improved algorithm was significantly better than other methods.

This paper is divided into five sections. Section 2 briefly describes the CLPSO algorithm and GWO algorithm. Section 3 provides a detailed definition of the GCLPSO. Section 4 is the experimental part, which details the experimental results of the GCLPSO and other comparison algorithms on these benchmark functions, feature selection, and engineering problems. Section 5 summarizes the contributions of this paper and plans for future work.

2. Background Knowledge

In this paper, the idea of the grey wolf algorithm is integrated into the CLPSO to strengthen the capability of the algorithm scouting for the optimal solution. This section will explain the grey wolf algorithm and the CLPSO in detail.

2.1. Particle Swarm Optimizer with Comprehensive Learning. Liang et al. proposed the CLPSO [61] algorithm in 2006. It uses a new comprehensive learning strategy (CLS) that uses the personal best position of the particle, *pbest*, to update the speed of the particle. The CLS can maintain the diversification of the population and prevent premature fall into a local optimum. The formula of speed and position update in the CLPSO algorithm is given as follows:

$$v_{id} = w * v_{id} + c * r_{id}\left(p\,\text{best}_{f_i(d),d} - x_{id}\right), \tag{1}$$

$$x_{id} = x_{id} + v_{id}, \tag{2}$$

where $f_i(d)$ represents the value of the dth dimension in a particle *pbest*, $f_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ represents the learning sample vector defined for particle i, and $\text{pbest}_{f_i(d),d}$ represents the optimal position of all particles *pbest* with the corresponding dimension value. The particle speed is updated by learning which dimension depends on the parameter learning probability *Pc*. When a dimension of a particle requires to update the speed, it will produce a random number. The corresponding dimension value will be learned from its own *pbest* if the random number is greater than *Pc*. Otherwise, it will learn from other particles *pbest*. The algorithm selects the learning particle from other particles as follows:

(1) Firstly, select two particles randomly from the population, excluding the particles that have updated the speed.

(2) Compare the fitness values *pbest* between the two particles and choose the best one. In this paper, the fitness value is the smallest solution of the function, which indicates that the function value is extremely small when solving the minimization problem.

CLPSO allocates a learning probability *Pc* to each particle by the following equation:

$$Pc_i = a + b \times \frac{\exp(10 \times (i-1)/N - 1)}{\exp(10) - 1}, \tag{3}$$

where a and b determine the maximum and minimum learning probabilities and N is the total number of particles.

Also, in order to avoid wasting time in bad directions when learning the optimal personal position of particles from samples, the threshold m of particle learning times was set. If the adaptive value of the particle is not improved after m times of continuous movement, a random particle will be generated to replace the particle. Pseudo-code of the particle f_i generation method in CLPSO is shown in Algorithm 1.

2.2. Grey Wolf Optimizer. Mirjalili et al. [62] proposed a new MA named GWO in 2014. The algorithm is inspired by the social level and hunting strategy of the wild grey wolf. In the GWO, the population is divided into four levels, including the highest alpha (α), beta (β), delta (δ), and the lowest omega (ω). The better wolves α, β, and δ lead the other wolves ω to explore the preferable solution field. In the GWO, wolves can spot the position of prey and encircle them.

$$\begin{aligned}\vec{D} &= \left|\vec{C} \times \vec{X}_p(t) - \vec{X}(t)\right|,\\ \vec{X}(t+1) &= \vec{X}_p(t) - \vec{A} * \vec{D},\end{aligned} \tag{4}$$

where $\vec{X}$ is the position vector of the grey wolf; $\vec{A}$ and $\vec{C}$ are coefficient vectors; $\vec{X}_p$ is the position vector of the prey; and t is the number of iterations.

The calculation method of $\vec{C}$ and $\vec{A}$ is shown as follows:

$$\begin{aligned}\vec{A} &= 2\vec{a} * \vec{r}_1 - \vec{a},\\ \vec{C} &= 2\vec{r}_2,\end{aligned} \tag{5}$$

where $\vec{r}_1$ and $\vec{r}_2$ are random numbers between [0, 1]; $\vec{a}$ decreases from 2 to 0 as the number of iterations increases. The hunting process of the grey wolf is shown by the following formula:

$$\vec{D}_\alpha = \left|\vec{C}_1 \times \vec{X}_\alpha - \vec{X}\right|, \tag{6}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \times \vec{X}_\beta - \vec{X}\right|, \tag{7}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \times \vec{X}_\delta - \vec{X}\right|, \tag{8}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \times \left(\vec{D}_\alpha\right), \tag{9}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \times \left(\vec{D}_\beta\right), \tag{10}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \times \left(\vec{D}_\delta\right), \tag{11}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{12}$$

The pseudo-code of the GWO is shown in Algorithm 2.

3. Proposed GCLPSO Method

This section interprets the GCLPSO in detail. In this paper, due to the slow convergence speed and low convergence accuracy of the CLPSO algorithm when dealing with optimization problems, core mechanisms of the GWO algorithm are cooperated to enhance the CLPSO algorithm, and a new algorithm called GCLPSO is proposed. The GWO's mechanisms can effectively promote the exploitative engine of the algorithm.

CLPSO updates the particle speed through *pbest* of all particles to avoid the algorithm from trapping in the local optimal solution prematurely and prevents the algorithm from carrying out local search near the global optimal solution. The elite-based dominance idea of the improved algorithm, called GCLPSO in this paper, is to select the three optimal solutions generated in each iteration of the CLPSO algorithm, such as GWO's alpha, beta, and delta, and then explore the vicinity of the three optimal solutions generated by the CLPSO algorithm through the GWO algorithm's idea. Meanwhile, the optimum searched is compared with the

Input the best position of each particle *pbest* and the fitness of an individual's best positions *fit* (*pbest*); *learn probability* $Pc_i\,(i = 1, 2, \ldots, N)$
for $d = 1 : D$
  if rand $\geq Pc_i$
    $f_i(d) = i$
  else select two particles a and b randomly $(a \neq b \neq i)$
    if $\text{fit}(\text{pbest}_a) < \text{fit}(\text{pbest}_b)$
      $f_i(d) = a$
    else
      $f_i(d) = b$
    end if
  end if
end for
if $f_i(d) == i\,(d = 1, \ldots, D)$in all dimensions
  select a particle randomly $j\,(j \neq i)$
  select a dimension d randomly
  $f_i(d) = j$
end if

Algorithm 1: Generation method for learning sample vector f_i.

Initialize the grey wolf population $X_i\,(i = 1, \ldots, N)$ and the parameters a, A, C
Calculate the fitness of each wolf
$X_\alpha =$ *the best wolf* α
$X_\beta =$ the second best wolf β
$X_\delta =$ the third best wolf δ
while $(t < T)//T$ is the maximum number of iterations
  for $i = 1 : N$
    Update the position of the current wolf by equation (12)
  end for
  Update a, A, and C
  Update X_α, X_β, and X_δ
  $t = t + 1$
end while
Return X_α

Algorithm 2: Pseudo-code of the GWO.

optimal solution in the CLPSO algorithm. If the optimum searched is superior to the optimum in the CLPSO algorithm, the optimal value of the CLPSO algorithm is updated to the global optimal solution. The improved algorithm enhances the local search of the CLPSO algorithm through the idea of the GWO algorithm. It boosts the local search capacity of the algorithm and enhances the accuracy and the acceleration of the algorithm under the condition that the algorithm does not trap in the local optimal solution in premature. The specific steps of the algorithm are described as follows:

(1) First, initialize the particles and parameters, and calculate the fitness for each particle.
(2) Update every particle with the CLPSO algorithm.
(3) The three optimal solutions of the CLPSO are selected as grey wolf algorithm alpha, beta, and delta, and the GWO is used to search locally near the optimal solution. If the optimum found is superior to the optimal solution in CLPSO, the optimal solution in CLPSO is replaced.
(4) Keep repeating Steps 2 and 3 until the condition of termination is met.

The specific process of the GCLPSO is described in Algorithm 3, which shows each step involved in the algorithm in detail. In order to illustrate the process of the improved algorithm more clearly and intuitively, the flowchart of the GCLPSO algorithm is shown in Figure 1.

The time complexity of the GCLPSO depends on the algorithm search population initialization $O\,(n)$ and the grey wolf population initialized to $O\,(n)$; update the search particle position to $O(n \times d \times g)$, update the local search for all grey wolf positions $O(n \times d \times g)$, and sort the population fitness values as $O(n \times \log n \times g)$. n is the size of the population, d is the dimension, and g is the maximum times of iterations. Therefore, the final time complexity of the GCLPSO algorithm is $2O(n \times d \times g + n) + O(n \times \log n \times g)$.

```
Set the maximum number of iterations, the threshold m, and the dimensionality of the space.
Generate learning sample vectors f_i using Algorithm 1; flag(i) = 0(i = 1, 2, ..., n)
Randomly generate the grey wolf population M_i (i = 1, ..., n); get the fitness of each agent fit(M_i)
Initialize a, A, and C
Create the initial population x_i (i = 1, ..., n); calculate the objective function value of x_i: fit(x_i)
Record the best position of each particle pbest and fitness of personal best positions fit(pbest); calculate the learning probability
Pc_i (i = 1, 2, ..., n)
l = 1;
while l < T
  for i = 1 : n
    if flag(i) > m
      generation learning sample vectors f_i using Algorithm 1
      flag(i) = 0
    end if
    Updating velocities and locations using equations (1) and (2)
    Compute fitness of population x_i
    if fit(x_i) < fit(pbest_i)//Update particle i pbest
      pbest_i = x_i; flag(i) = 0
    else
      flag(i) = flag(i) + 1
    end if
  end for
  Update global optimal solution gbest
  Select the best three solutions x_a, x_b, and x_c from fit(x_i) as alpha M_α, beta M_β, and delta M_δ
  for i = 1 : n
    Update the position of the current wolf by equation (12)
  end for
  Update a, C, and A
  Calculate the fitness of each search agent; select the best solution M_a
  If fit(gbest) > fit(M_a)
    gbest = M_a
    x_a = M_a
  end if
end while
Return gbest
```

Algorithm 3: Pseudo-code of the GCLPSO.

4. Experimental Studies

This section firstly compares the algorithm proposed in this paper with other advanced methods through 30 classic benchmark functions in CEC2017. The performance of the GCLPSO algorithm on benchmark functions was verified. Then, the algorithm was applied to the design of three engineering construction problems, and good optimization results were obtained, which confirms the ability of the algorithm in coping with constraints.

4.1. Benchmarks' Validation. In this paper, 30 classical benchmarks in CEC2017 were utilized to compare the algorithm proposed in this paper with other advanced methods. These functions consist of unimodal (C01–C03), multimodal (C04–C10), hybrid (C11–C20), and composition functions (C21–C30). The performance of the algorithm was evaluated more comprehensively by different types of benchmarks. The descriptions of the 30 benchmarks are shown in Table 1. The CLPSO is also compared with other PSO variants on 10 classic benchmark functions in CEC2019. The benchmark functions of CEC2019 are shown in Table 2.

To obtain fair and unbiased results, the experiment was carried out with the same parameter setting: the population size and the maximum number of iterations were set to 30 and 2000, accordingly. Each competitor runs independently thirty times on the benchmark functions. Then, the Friedman test [80] is used to comprehensively assess the optimal results of all competitors on the benchmarks. The Friedman test is a nonparametric statistical comparison test, which is usually adopted to distinguish the differences between multiple test results. Then, the average performance of the selected method is sorted, and further statistical comparison is carried out to achieve the ARV (average sort value) in the result of comparison. Moreover, the paired Wilcoxon symbolic rank test [81] was adopted for the statistical test to detect the significant difference between the two sample mean values. Only in the condition that the p value obtained was less than 0.05, the performance of the GCLPSO was considered to be significantly superior to other competitors. In this paper, two effective test approaches were applied to

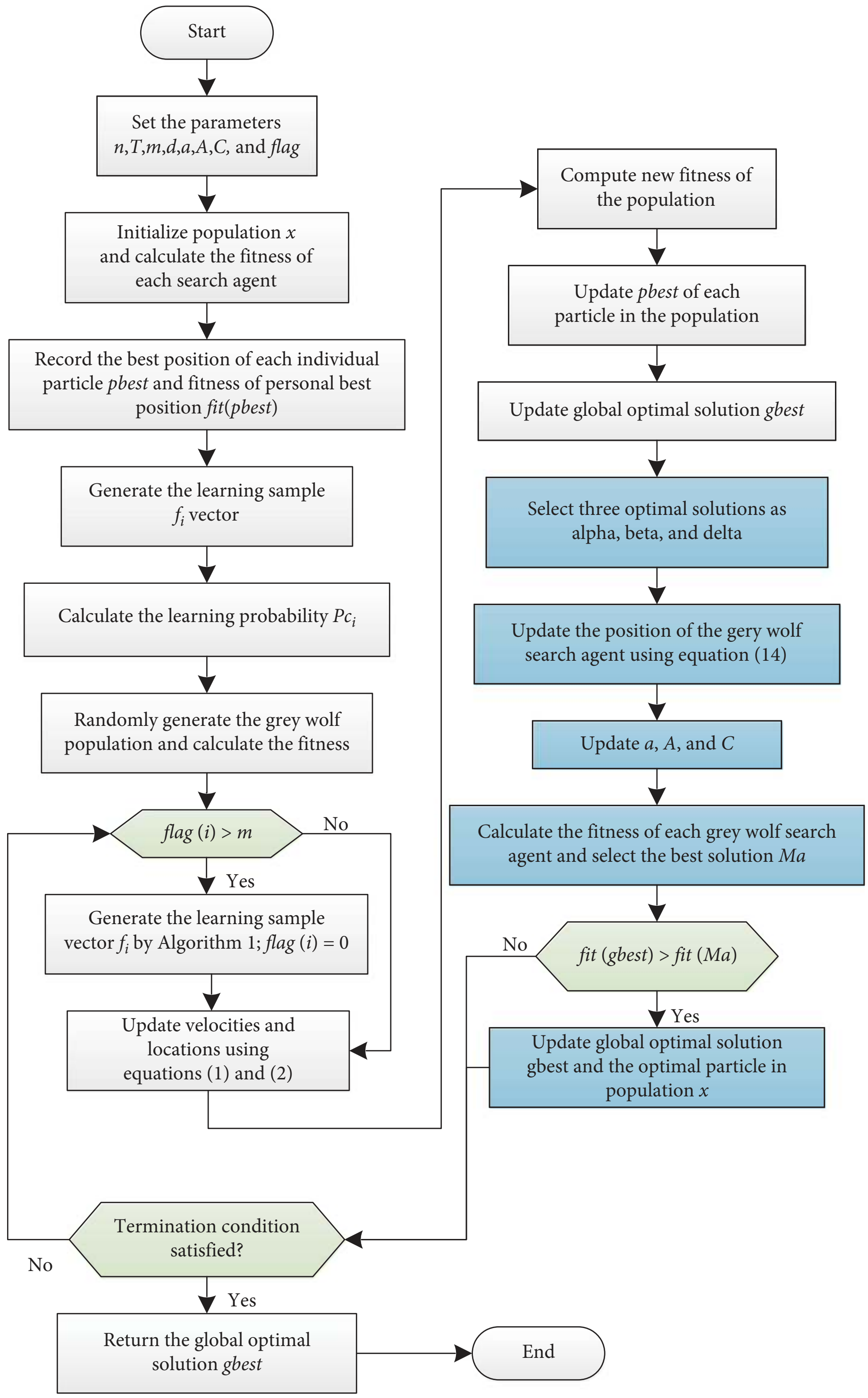

Figure 1: Flowchart of the GCLPSO.

compare the advantages and disadvantages of different algorithms tested on 30 benchmarks in CEC2017 and verify the effect of the algorithm.

4.2. Comparisons of the GCLPSO with Other Algorithms. An this section, several MAs used were compared with the GCLPSO on 30 benchmark functions in CEC2017. In order to fully certify the performance of the GCLPSO, this paper uses seven classical MAs and eight advanced MAs as comparison algorithms. The classical MAs involved are as follows: PSO [45], GOA [82], DA [71], MFO [29], SCA [72], WOA [24], and GWO [62]. The advanced MAs include CGSCA [73], SCADE [74], CIFOA [75], AMFOA [76], LWOA [77], IWOA [78], BLPSO [79], and CLPSO [61]. As shown in Table 3, the parameters of all

Table 1: CEC2017 test functions.

Function	Function name	Optimum
CEC2017 unimodal functions (UF)		
C01	Shifted and rotated bent cigar function	100
C02	Shifted and rotated sum of different power function	200
C03	Shifted and rotated Zakharov function	300
CEC2017 multimodal functions (MF)		
C04	Shifted and rotated Rosenbrock's function	400
C05	Shifted and rotated Rastrigin's function	500
C06	Shifted and rotated expanded Scaffer's F6 function	600
C07	Shifted and rotated Lunacek bi-Rastrigin function	700
C08	Shifted and rotated noncontinuous Rastrigin's function	800
C09	Shifted and rotated Lévy function	900
C10	Shifted and rotated Schwefel's function	1000
CEC2017 hybrid functions (HF)		
C11	HF 1 ($N=3$)	1100
C12	HF 2 ($N=3$)	1200
C13	HF 3 ($N=3$)	1300
C14	HF 4 ($N=4$)	1400
C15	HF 5 ($N=4$)	1500
C16	HF 6 ($N=4$)	1600
C17	HF 6 ($N=5$)	1700
C18	HF 6 ($N=5$)	1800
C19	HF 6 ($N=5$)	1900
C20	HF 6 ($N=6$)	2000
CEC2017 composition functions (CF)		
C21	CF 1 ($N=3$)	2100
C22	CF 2 ($N=3$)	2200
C23	CF 3 ($N=4$)	2300
C24	CF 4 ($N=4$)	2400
C25	CF 5 ($N=5$)	2500
C26	CF 6 ($N=5$)	2600
C27	CF 7 ($N=6$)	2700
C28	CF 8 ($N=6$)	2800
C29	CF 9 ($N=3$)	2900
C30	CF 10 ($N=3$)	3000

Table 2: CEC2019 test functions.

Function	Name of the function	$\mathbf{F_i^*} = \mathbf{F_i}(\mathbf{X}^*)$	D	Search range
C31	Storn's Chebyshev polynomial fitting problem	1	9	[−8192, 8192]
C32	Inverse Hilbert matrix problem	1	16	[−16834, 16834]
C33	Lennard-Jones minimum energy cluster	1	18	[−4, 4]
C34	Rastrigin's function	1	10	[−100, 100]
C35	Griewank's function	1	10	[−100, 100]
C36	Weierstrass function	1	10	[−100, 100]
C37	Modified Schwefel's function	1	10	[−100, 100]
C38	Expanded Schaffer's F6 function	1	10	[−100, 100]
C39	Happy Cat function	1	10	[−100, 100]
C40	Ackley function	1	10	[−100, 100]

comparison algorithms were set according to the original paper. In order to make the experimental results fair and reliable, all algorithms were executed in the same environment.

In the experiment, the maximum number of iterations is set to 2000, and the population size is set to 30. Each algorithm runs 30 times independently on the benchmark function. The comparison results are shown in Table 4, where the mean and standard deviation of the algorithm after 30 independent executions on 30 benchmark functions are listed. On the unimodal benchmark functions, LWOA and PSO in the C2 case have strong optimization ability, and the optimum of the LWOA algorithm is superior to all other competitors. The PSO algorithm in the C3 case has strong competitiveness, and its final result is superior to other algorithms. In the case of C1 and C4, the optimization result

Table 3: Parameter setting for compared algorithms.

Method	Maximum generation	Population size	Other parameters
GCLPSO	2000	30	$m=5$; $a\in[0\ 2]$
PSO	2000	30	$c_1=2$; $c_2=2$; $vMax=6$
MFO	2000	30	$b=1$; $t=[-1\ 1]$; $a\in[-2\text{–}1]$
GOA	2000	30	$c\text{Max}=1$; $c\text{Min}=0.00004$
DA	2000	30	$w\in[0.2\ 0.9]$; $s=0.1$; $a=0.1$; $c=0.7$; $f=1$; $e=1$
SCA	2000	30	$A=2$
WOA	2000	30	$a_1=[2\ 0]$; $a_2=[-2\text{–}1]$; $b=1$
GWO	2000	30	$a=[2,\ 0]$
CLPSO	2000	30	$w=[0.2\ 0.9]$; $c=1.496$
BLPSO	2000	30	$w=[0.2\ 0.9]$; $c=1.496$; $I=1$; $E=1$
SCADE	2000	40	$P_{CR}=0.2$
CGSCA	2000	40	$a=2$
CIFOA	2000	30	$Mr=0.8$
AMFOA	2000	30	$FR\in[-10,\ 10]$
IWOA	2000	30	$a_1=[2\ 0]$; $a_2=[-2\text{–}1]$; $b=1$
LWOA	2000	30	$\beta=1.5$; $l\in[-1,\ 1]$; $b=1$

Table 4: Comparison of results of different algorithms.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
GCLPSO	**5.27E+03**	**5.17E+03**	**3.38E+17**	**1.03E+18**	**6.56E+03**	**3.17E+03**
CLPSO	1.80E+07	5.96E+06	6.21E+25	1.76E+26	9.83E+04	1.66E+04
BLPSO	3.76E+08	1.05E+08	1.03E+27	2.54E+27	6.09E+04	1.27E+04
IWOA	1.63E+08	2.30E+08	3.19E+30	1.50E+31	1.88E+05	7.07E+04
LWOA	1.18E+06	3.26E+05	**4.63E+10**	**1.45E+11**	1.87E+04	1.38E+04
AMFOA	8.00E+10	4.40E+09	6.96E+59	1.10E+60	8.35E+08	4.68E+08
CIFOA	7.74E+10	4.14E+08	2.15E+55	6.75E+55	8.91E+04	6.41E+02
SCADE	2.29E+10	3.12E+09	6.32E+37	2.31E+38	6.58E+04	5.88E+03
CGSCA	1.77E+10	3.11E+09	7.17E+36	1.95E+37	5.71E+04	7.93E+03
GWO	2.63E+09	2.02E+09	1.26E+31	3.23E+31	4.46E+04	9.63E+03
WOA	3.47E+08	2.03E+08	1.27E+36	6.96E+36	2.57E+05	7.57E+04
MFO	9.34E+09	7.94E+09	1.84E+43	1.01E+44	1.25E+05	5.44E+04
SCA	1.62E+10	2.96E+09	9.78E+36	4.25E+37	5.58E+04	9.14E+03
GOA	1.25E+06	1.61E+06	1.22E+27	6.70E+27	6.88E+03	3.38E+03
DA	2.52E+09	2.06E+09	5.96E+37	1.97E+38	1.04E+05	2.52E+04
PSO	1.59E+08	2.52E+07	**2.31E+14**	**2.79E+14**	**2.35E+03**	**1.05E+03**
	C4		C5		C6	
GCLPSO	**4.37E+02**	**3.16E+01**	**6.32E+02**	**2.66E+01**	**6.02E+02**	**1.01E+00**
CLPSO	5.76E+02	2.13E+01	6.54E+02	1.85E+01	6.04E+02	7.76E−01
BLPSO	6.43E+02	2.91E+01	7.15E+02	1.68E+01	6.14E+02	1.61E+00
IWOA	5.95E+02	5.18E+01	7.98E+02	6.50E+01	6.61E+02	7.36E+00
LWOA	5.07E+02	2.54E+01	7.72E+02	6.30E+01	6.63E+02	1.25E+01
AMFOA	2.58E+04	1.98E+03	1.07E+03	3.35E+01	7.29E+02	4.72E+00
CIFOA	3.05E+04	3.05E+02	1.01E+03	8.65E+00	7.05E+02	2.33E+00
SCADE	4.59E+03	1.11E+03	8.42E+02	2.57E+01	6.69E+02	7.80E+00
CGSCA	2.65E+03	8.69E+02	8.20E+02	2.38E+01	6.63E+02	7.17E+00
GWO	6.15E+02	9.11E+01	**6.09E+02**	**2.42E+01**	6.10E+02	4.94E+00
WOA	6.85E+02	6.57E+01	8.05E+02	5.55E+01	6.79E+02	1.71E+01
MFO	1.20E+03	8.37E+02	7.04E+02	4.60E+01	6.40E+02	1.05E+01
SCA	2.14E+03	5.67E+02	8.00E+02	1.86E+01	6.56E+02	6.48E+00
GOA	5.13E+02	2.36E+01	6.45E+02	3.27E+01	6.46E+02	1.89E+01
DA	1.22E+03	6.47E+02	8.74E+02	8.02E+01	6.76E+02	1.24E+01
PSO	4.75E+02	3.71E+01	7.55E+02	2.80E+01	6.55E+02	1.19E+01
	C7		C8		C9	
GCLPSO	**8.31E+02**	**3.98E+01**	**8.98E+02**	**4.01E+01**	**1.61E+03**	**5.56E+02**
CLPSO	9.14E+02	1.60E+01	9.69E+02	1.64E+01	2.90E+03	8.88E+02
BLPSO	1.01E+03	2.02E+01	1.02E+03	1.24E+01	1.81E+03	2.07E+02

Table 4: Continued.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
IWOA	1.20E+03	9.06E+01	9.98E+02	3.63E+01	9.28E+03	4.13E+03
LWOA	1.12E+03	7.80E+01	9.93E+02	3.54E+01	7.95E+03	3.01E+03
AMFOA	1.58E+03	2.79E+01	1.26E+03	1.84E+01	1.68E+04	2.98E+03
CIFOA	1.48E+03	1.09E+01	1.21E+03	8.19E+00	1.17E+04	7.72E+02
SCADE	1.22E+03	3.99E+01	1.09E+03	2.71E+01	9.09E+03	1.07E+03
CGSCA	1.21E+03	4.70E+01	1.08E+03	1.79E+01	7.99E+03	1.40E+03
GWO	8.75E+02	3.69E+01	**8.92E+02**	**2.28E+01**	2.00E+03	6.29E+02
WOA	1.28E+03	7.99E+01	1.01E+03	4.03E+01	9.69E+03	4.06E+03
MFO	1.14E+03	2.33E+02	1.01E+03	3.90E+01	7.53E+03	2.63E+03
SCA	1.19E+03	4.73E+01	1.07E+03	1.93E+01	6.48E+03	1.41E+03
GOA	8.83E+02	5.38E+01	9.38E+02	3.76E+01	5.53E+03	4.24E+03
DA	1.07E+03	7.23E+01	1.10E+03	5.14E+01	1.32E+04	4.72E+03
PSO	9.46E+02	2.07E+01	1.01E+03	2.11E+01	6.94E+03	2.26E+03
	C10		C11		C12	
GCLPSO	**5.34E+03**	**4.52E+02**	**1.22E+03**	**5.31E+01**	**6.81E+05**	**2.94E+05**
CLPSO	6.45E+03	3.78E+02	1.46E+03	8.52E+01	1.94E+07	5.10E+06
BLPSO	8.16E+03	3.71E+02	1.45E+03	4.43E+01	3.76E+07	9.67E+06
IWOA	6.00E+03	6.15E+02	2.56E+03	1.17E+03	4.48E+07	3.99E+07
LWOA	5.70E+03	7.76E+02	1.28E+03	6.76E+01	6.08E+06	3.76E+06
AMFOA	9.92E+03	2.76E+02	5.53E+08	1.74E+08	2.59E+10	8.72E+08
CIFOA	8.93E+03	1.55E+02	2.04E+05	2.66E+05	2.62E+10	1.51E+09
SCADE	8.48E+03	2.79E+02	4.07E+03	6.84E+02	2.54E+09	6.61E+08
CGSCA	8.68E+03	2.46E+02	2.97E+03	5.57E+02	1.97E+09	5.60E+08
GWO	4.19E+03	6.35E+02	2.25E+03	1.09E+03	6.01E+07	7.49E+07
WOA	6.91E+03	8.49E+02	4.33E+03	1.72E+03	1.31E+08	1.09E+08
MFO	5.54E+03	6.87E+02	5.06E+03	4.25E+03	5.43E+08	9.58E+08
SCA	8.58E+03	3.79E+02	2.96E+03	7.73E+02	1.74E+09	3.61E+08
GOA	5.26E+03	8.68E+02	1.39E+03	9.03E+01	1.99E+07	2.25E+07
DA	7.09E+03	7.87E+02	2.73E+03	9.50E+02	5.70E+08	5.21E+08
PSO	6.49E+03	5.07E+02	1.31E+03	3.35E+01	3.56E+07	1.95E+07
	C13		C14		C15	
GCLPSO	**2.58E+04**	**1.50E+04**	**7.43E+03**	**2.88E+03**	**6.94E+03**	**6.29E+03**
CLPSO	4.45E+06	4.30E+06	8.59E+04	8.47E+04	1.31E+05	1.81E+05
BLPSO	2.82E+06	2.57E+06	9.32E+04	6.71E+04	4.07E+05	3.19E+05
IWOA	2.18E+05	3.24E+05	1.38E+06	1.17E+06	3.61E+04	2.53E+04
LWOA	1.88E+05	1.12E+05	3.83E+04	3.15E+04	7.44E+04	3.98E+04
AMFOA	3.76E+10	1.81E+09	9.39E+08	3.47E+08	2.19E+09	1.04E+09
CIFOA	3.49E+10	7.06E+09	1.59E+08	1.42E+08	4.32E+09	3.26E+08
SCADE	8.99E+08	3.20E+08	5.89E+05	4.78E+05	1.28E+07	1.43E+07
CGSCA	8.57E+08	3.24E+08	4.65E+05	3.71E+05	2.56E+07	2.86E+07
GWO	2.20E+07	8.58E+07	4.15E+05	4.51E+05	2.07E+06	7.77E+06
WOA	3.84E+05	4.25E+05	1.77E+06	1.49E+06	2.75E+05	2.92E+05
MFO	1.29E+07	2.68E+07	4.87E+05	1.34E+06	6.07E+04	4.87E+04
SCA	7.10E+08	3.76E+08	3.21E+05	1.89E+05	2.84E+07	2.64E+07
GOA	1.65E+05	1.35E+05	3.31E+04	3.19E+04	7.36E+04	4.72E+04
DA	7.80E+07	1.97E+08	1.15E+06	2.03E+06	2.27E+05	3.24E+05
PSO	7.72E+06	2.01E+06	2.40E+04	1.69E+04	9.43 E+05	2.78E+05
	C16		C17		C18	
GCLPSO	**2.65E+03**	**2.21E+02**	**1.87E+03**	**7.69E+01**	**2.15E+05**	**1.55E+05**
CLPSO	**2.65E+03**	**2.05E+02**	2.01E+03	8.43E+01	7.26E+05	4.27E+05
BLPSO	3.16E+03	2.22E+02	2.08E+03	1.01E+02	1.87E+06	7.97E+05
IWOA	3.20E+03	3.97E+02	2.52E+03	2.58E+02	3.30E+06	3.63E+06
LWOA	3.13E+03	3.45E+02	2.45E+03	2.86E+02	7.07E+05	5.97E+05
AMFOA	2.70E+04	9.04E+02	1.16E+05	2.37E+04	2.15E+09	8.74E+08
CIFOA	1.20E+04	3.61E+03	8.42E+04	5.51E+04	1.46E+09	8.96E+08
SCADE	4.09E+03	2.24E+02	2.61E+03	1.32E+02	6.18E+06	3.45E+06
CGSCA	3.98E+03	2.38E+02	2.68E+03	1.77E+02	8.84E+06	4.94E+06
GWO	2.56E+03	2.95E+02	1.99E+03	1.44E+02	1.10E+06	1.25E+06
WOA	4.01E+03	8.36E+02	2.64E+03	2.82E+02	5.29E+06	4.96E+06

Table 4: Continued.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
MFO	3.05E+03	4.39E+02	2.50E+03	2.56E+02	2.11E+06	3.06E+06
SCA	3.95E+03	1.55E+02	2.61E+03	1.68E+02	8.02E+06	5.96E+06
GOA	2.88E+03	3.64E+02	2.25E+03	2.01E+02	7.01E+05	1.05E+06
DA	4.00E+03	6.16E+02	2.89E+03	3.52E+02	6.98E+06	8.23E+06
PSO	3.01E+03	2.40E+02	2.40E+03	2.54E+02	4.30E+05	2.93E+05
	C19		C20		C21	
GCLPSO	**4.90E+03**	**2.84E+03**	**2.36E+03**	**1.02E+02**	**2.41E+03**	**4.74E+01**
CLPSO	7.96E+04	7.00E+04	2.37E+03	1.04E+02	2.45E+03	3.50E+01
BLPSO	5.11E+05	4.42E+05	2.44E+03	9.95E+01	2.51E+03	1.61E+01
IWOA	2.43E+05	4.53E+05	2.70E+03	2.25E+02	2.58E+03	6.61E+01
LWOA	4.31E+05	2.90E+05	2.78E+03	2.29E+02	2.56E+03	7.41E+01
AMFOA	3.27E+09	6.38E+08	4.14E+03	1.71E+02	3.08E+03	4.91E+01
CIFOA	4.56E+09	1.10E+08	3.97E+03	1.69E+02	3.01E+03	5.77E+01
SCADE	4.91E+07	3.43E+07	2.82E+03	1.13E+02	2.60E+03	2.41E+01
CGSCA	6.15E+07	2.75E+07	2.75E+03	1.33E+02	2.60E+03	2.40E+01
GWO	9.45E+05	1.26E+06	2.43E+03	1.37E+02	2.39E+03	2.34E+01
WOA	8.79E+06	9.13E+06	2.82E+03	2.05E+02	2.60E+03	4.99E+01
MFO	1.59E+07	3.80E+07	2.72E+03	2.24E+02	2.50E+03	5.32E+01
SCA	5.09E+07	3.07E+07	2.77E+03	1.44E+02	2.58E+03	2.90E+01
GOA	3.65E+06	3.12E+06	2.60E+03	1.77E+02	2.43E+03	3.20E+01
DA	4.87E+07	6.38E+07	2.83E+03	1.90E+02	2.66E+03	7.42E+01
PSO	2.66E+06	1.27E+06	2.72E+03	1.71E+02	2.56E+03	3.67E+01
	C22		C23		C24	
GCLPSO	**2.30E+03**	**1.41E+00**	**2.73E+03**	**3.30E+01**	**2.91E+03**	**3.13E+01**
CLPSO	3.84E+03	1.88E+03	2.81E+03	2.13E+01	3.00E+03	7.19E+01
BLPSO	2.40E+03	1.36E+01	2.87E+03	1.88E+01	3.05E+03	1.47E+01
IWOA	6.52E+03	2.21E+03	3.04E+03	8.27E+01	3.16E+03	7.66E+01
LWOA	6.16E+03	1.64E+03	3.00E+03	7.99E+01	3.20E+03	1.17E+02
AMFOA	1.22E+04	3.83E+02	6.48E+03	6.15E+02	5.05E+03	5.20E+01
CIFOA	1.13E+04	1.41E+02	4.09E+03	2.34E+02	4.30E+03	3.51E+02
SCADE	5.13E+03	7.57E+02	3.03E+03	3.83E+01	3.20E+03	3.73E+01
CGSCA	4.73E+03	1.54E+03	3.04E+03	5.07E+01	3.19E+03	2.80E+01
GWO	4.94E+03	1.80E+03	2.76E+03	3.61E+01	2.94E+03	5.35E+01
WOA	7.89E+03	1.03E+03	3.12E+03	1.01E+02	3.19E+03	1.07E+02
MFO	6.86E+03	8.83E+02	2.84E+03	4.03E+01	2.99E+03	3.20E+01
SCA	9.36E+03	1.77E+03	3.04E+03	3.46E+01	3.20E+03	3.33E+01
GOA	5.75E+03	1.86E+03	2.80E+03	3.99E+01	2.96E+03	3.83E+01
DA	7.65E+03	2.10E+03	3.28E+03	1.75E+02	3.45E+03	1.62E+02
PSO	5.56E+03	2.89E+03	3.14E+03	1.04E+02	3.24E+03	1.18E+02
	C25		C26		C27	
GCLPSO	**2.88E+03**	**8.68E+00**	**4.37E+03**	**9.09E+02**	**3.20E+03**	**2.04E−04**
CLPSO	2.93E+03	1.08E+01	4.71E+03	6.61E+02	3.25E+03	9.38E+00
BLPSO	2.97E+03	1.81E+01	5.79E+03	5.24E+02	3.30E+03	1.30E+01
IWOA	2.99E+03	3.27E+01	7.31E+03	1.32E+03	3.34E+03	8.26E+01
LWOA	2.91E+03	2.02E+01	6.62E+03	1.41E+03	3.28E+03	4.33E+01
AMFOA	7.56E+03	3.73E+02	1.48E+04	6.94E+02	9.19E+03	4.95E+02
CIFOA	7.92E+03	6.90E+01	1.47E+04	3.21E+02	5.67E+03	7.11E+02
SCADE	3.57E+03	1.56E+02	7.76E+03	3.46E+02	3.50E+03	5.71E+01
CGSCA	3.43E+03	1.81E+02	7.56E+03	5.63E+02	3.47E+03	5.82E+01
GWO	2.98E+03	3.83E+01	4.84E+03	3.82E+02	3.26E+03	2.09E+01
WOA	3.05E+03	4.43E+01	7.38E+03	1.38E+03	3.45E+03	1.13E+02
MFO	3.31E+03	5.67E+02	5.81E+03	4.31E+02	3.26E+03	3.22E+01
SCA	3.40E+03	1.37E+02	7.30E+03	2.98E+02	3.47E+03	7.13E+01
GOA	2.93E+03	2.87E+01	5.29E+03	9.69E+02	3.24E+03	1.99E+01
DA	3.17E+03	1.82E+02	8.66E+03	1.60E+03	3.56E+03	1.72E+02
PSO	2.92E+03	2.79E+01	5.52E+03	1.98E+03	3.20E+03	7.86E+01
	C28		C29		C30	
GCLPSO	3.29E+03	5.07E+01	3.62E+03	1.12E+02	1.25E+04	3.19E+03
CLPSO	3.36E+03	2.68E+01	3.88E+03	1.31E+02	8.61E+05	5.60E+05
BLPSO	3.35E+03	1.68E+01	4.11E+03	1.31E+02	1.98E+06	1.05E+06

Table 4: Continued.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
IWOA	3.39 $E+03$	4.14$E+01$	4.66$E+03$	4.04$E+02$	3.13$E+06$	2.03$E+06$
LWOA	3.23$E+03$	2.62$E+01$	4.33$E+03$	3.27$E+02$	1.46$E+06$	8.15$E+05$
AMFOA	8.81$E+03$	3.17$E+02$	1.44$E+05$	3.35$E+04$	8.57$E+09$	5.14$E+08$
CIFOA	9.49$E+03$	4.61$E+01$	6.93$E+04$	3.77$E+04$	5.77$E+09$	2.31$E+09$
SCADE	4.53$E+03$	3.99$E+02$	5.29$E+03$	2.40$E+02$	1.49$E+08$	4.94$E+07$
CGSCA	4.30$E+03$	3.14$E+02$	4.96$E+03$	2.36$E+02$	1.48$E+08$	6.50$E+07$
GWO	3.42$E+03$	7.98$E+01$	3.75$E+03$	1.49$E+02$	7.73$E+06$	7.45$E+06$
WOA	3.52$E+03$	5.51$E+02$	5.14$E+03$	5.68$E+02$	2.24$E+07$	2.14$E+07$
MFO	4.20$E+03$	9.42$E+02$	4.19$E+03$	3.11$E+02$	1.17$E+06$	2.04$E+06$
SCA	4.14$E+03$	2.39$E+02$	4.95$E+03$	2.26$E+02$	1.12$E+08$	4.56$E+07$
GOA	3.27$E+03$	2.84$E+01$	4.14$E+03$	1.90$E+02$	7.67$E+06$	5.76$E+06$
DA	3.80$E+03$	2.01$E+02$	5.45$E+03$	6.82$E+02$	3.89$E+07$	2.96$E+07$
PSO	3.27$E+03$	2.36$E+01$	4.39$E+03$	2.47$E+02$	5.73$E+06$	2.10$E+06$

	Overall rank Rank	ARV	+/=/−
GCLPSO	**1**	**1.755556**	27/3/0
CLPSO	4	4.804444	30/0/0
BLPSO	6	6.304444	29/1/0
IWOA	9	8.087778	30/0/0
LWOA	5	5.976667	30/0/0
AMFOA	16	15.68444	30/0/0
CIFOA	15	15.06667	30/0/0
SCADE	14	11.80667	30/0/0
CGSCA	13	11.28667	30/0/0
GWO	3	4.71	24/3/3
WOA	10	10.08778	30/0/0
MFO	8	7.354444	29/1/0
SCA	11	11.00444	30/0/0
GOA	2	4.537778	26/3/1
DA	12	11.14333	30/0/0
PSO	7	6.388889	27/0/3

of the algorithm in this paper is superior to all the competition algorithms. In the multimodal benchmark functions, C5 and C8 cases, the solution of the GWO algorithm is greater than that of the GCLPSO algorithm, but in the functions of C6, C7, C9, and C10, the optimal solution obtained by the GCLPSO algorithm is higher than that of all other competing algorithms. On hybrid benchmark functions, GCLPSO algorithm is better compared to other algorithms. In the composition benchmark functions, in addition to the strong competitive advantage of LWOA, GOA, and PSO in the C28 function, GCLPSO has obvious advantages in other functions. The GCLPSO algorithm uses the GWO algorithm to strengthen the local scout of the CLPSO algorithm so that the algorithm has a better harmony between local search and global search and possesses an optimal solution on most benchmark functions. This proves that the algorithm can effectively deal with unimodal, multimodal, hybrid, and composition functions at the same time.

Also, the Friedman test and Wilcoxon signed-rank test were used to evaluate the comprehensive effect of the algorithm. The Wilcoxon symbol rank test measures the p values of all comparison algorithms on 30 benchmark functions, which are basically less than 0.05. At the same time, the symbol "+/=/−" can indicate that the GCLPSO is significantly superior to 15 competing algorithms on 30 benchmarks of 4 different types. Table 4 also shows the Friedman test comparison results. GCLPSO has the lowest ARV among the 30 benchmark functions. It is proved that the GCLPSO algorithm is greater than other popular algorithms in CEC2017. Therefore, the algorithm proposed by us has a preferable convergence rate and a more accurate convergence solution than other competitors.

In order to more intuitively and clearly understand the convergence trend of the algorithm in terms of functions and estimate the performance of the algorithm, a representative benchmark function was selected from CEC2017 for analysis, and images demonstrated the convergence process of the algorithm. As shown in Figure 2, on unimodal benchmark function C1, the convergence tendency and precision of the GCLPSO algorithm are better than other algorithms. On multimodal benchmark functions C7 and C9, the convergence rate of the PSO in the early stage of the C7 function iteration is better than that of the GCLPSO algorithm, the convergence tendency of the GWO and BLPSO algorithm in the early stage of the C9 function iteration is greater than that of the GCLPSO algorithm, but the optimal value of the GCLPSO algorithm in the late stage of convergence is better than that of all other competing algorithms. On hybrid benchmark functions C12, C13, and C9, GCLPSO algorithm is lower than some

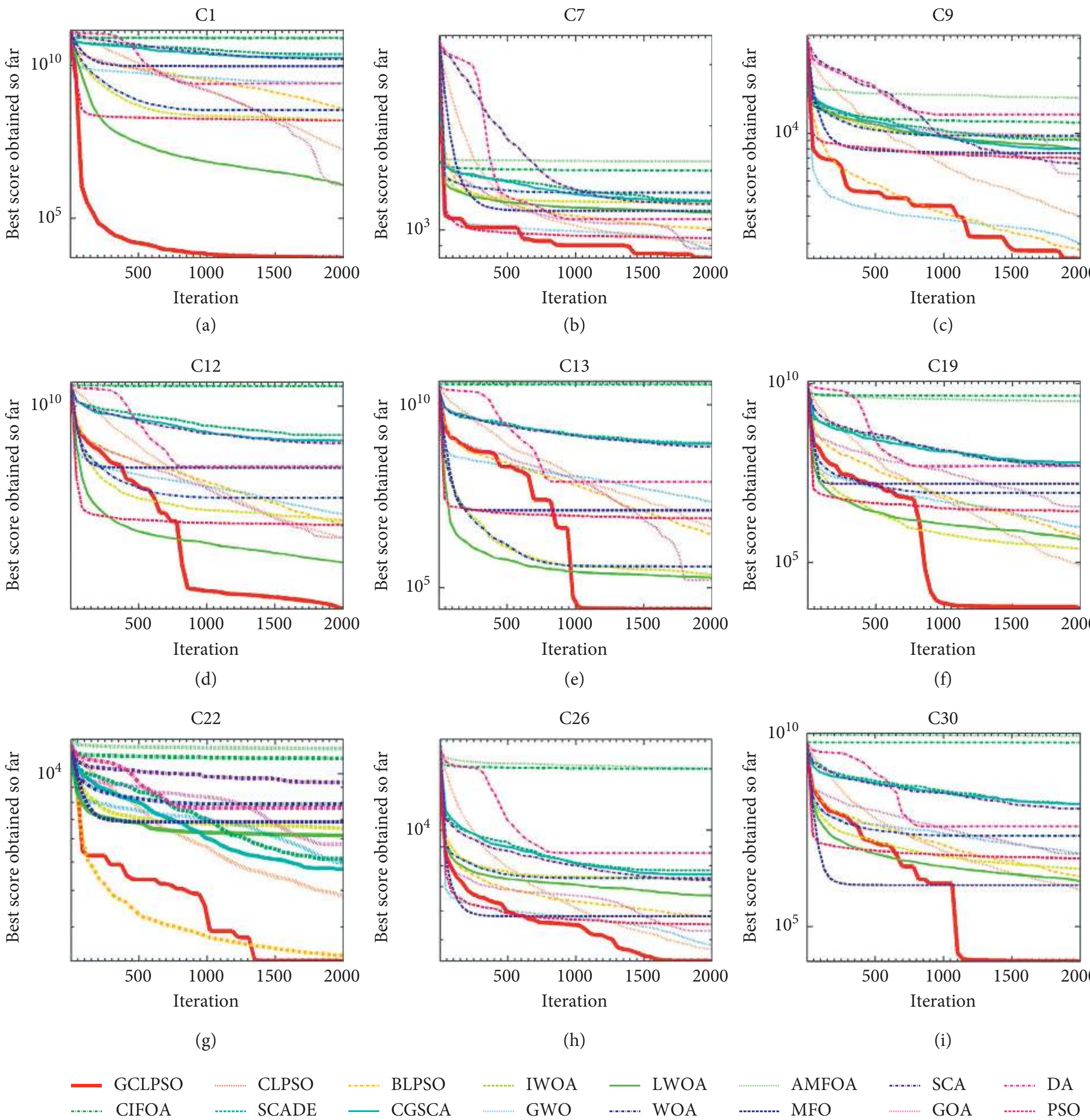

Figure 2: Convergence trend of the GCLPSO and other methods.

competitors in the early convergence period, but the optimal value of convergence in the late convergence period is much higher than other competing algorithms. On the composition benchmark functions C22, C26, and C30, although the BLPSO algorithm is highly competitive on function C22, the optimal values searched by the GCLPSO algorithm on the three benchmark functions are all higher than other algorithms.

4.3. Comparisons of the GCLPSO with Other PSO Variants. In this section, GCLPSO was compared with the improved PSO variants on the benchmark functions of CEC2017 and CEC2019. The advanced PSO variants include FST-PSO [83], PP-PSO [84], and SopPSO [85]. In order to make the experimental results fair and reliable, all algorithms were executed in the same environment.

In the experiment, the population size is set to 30, and the maximum number of iterations is set to 1000. Each algorithm runs independently on the benchmark function 30 times. The comparison results are shown in Table 5, which lists the average adaptive value and standard deviation of the algorithm after the algorithm is independently executed 30 times on 40 benchmark functions. The GCLPSO achieves optimal fitness values on most benchmark functions. As shown in Table 5, the optimal fitness value is shown in bold. On unimodal functions C1 and C3, the adaptive value searched by the GCLPSO is better than other comparison algorithms. On multimodal functions C5, C6, C7, C8, and

Table 5: Comparison of results for different algorithms.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
GCLPSO	**2.96E + 08**	4.76E + 08	1.08E + 30	1.9E + 30	**21413.4**	4508.552
FST-PSO	4.59E + 10	1.44E + 10	1.53E + 39	3.93E + 39	102540.3	31341.76
PP-PSO	4.91E + 08	1.23E + 09	1.74E + 27	5.34E + 27	37336.02	12931.16
SopPSO	69643742	1.37E + 08	**9.44E + 19**	1.41E + 20	33801.34	5568.606
	C4		C5		C6	
GCLPSO	725.2314	127.8274	**624.5366**	31.30305	**608.0114**	4.335014
FST-PSO	5249.981	1590.092	763.0822	32.33813	666.6076	10.16737
PP-PSO	**517.4305**	36.73232	692.7252	20.70859	655.9324	7.717476
SopPSO	569.4742	47.2971	694.8522	22.79289	631.6974	5.726442
	C7		C8		C9	
GCLPSO	**853.2442**	31.01812	**915.111**	47.39302	**2759.384**	729.8192
FST-PSO	1474.497	143.8618	1139.403	39.09915	8175.504	3166.286
PP-PSO	1246.102	73.45671	1058.999	55.08997	8181.35	2684.43
SopPSO	979.2978	25.34037	1023.224	38.57947	5550.586	2333.192
	C10		C11		C12	
GCLPSO	6902.995	1104.216	1546.228	397.4099	1.93E + 08	1.3E + 08
FST-PSO	7545.323	502.1786	15587.53	7038.758	4.86E + 09	2.3E + 09
PP-PSO	**6630.596**	706.92	**1448.767**	119.4996	**51630157**	39779490
SopPSO	6625.082	667.6307	1515.749	100.238	1.15E + 08	45378499
	C13		C14		C15	
GCLPSO	6311983	16366176	**18691.68**	14472.48	339311.4	771517.5
FST-PSO	3.29E + 08	2.46E + 08	513636	883827.9	3126763	8917858
PP-PSO	**113435.6**	70563.99	21442.57	37439.27	**17791.48**	10822.85
SopPSO	334284.1	185471	25319.19	28123.36	86743.07	67917.81
	C16		C17		C18	
GCLPSO	**2524.394**	349.3318	**2181.644**	214.8127	303575.8	508335.9
FST-PSO	3875.41	444.8716	2807.213	557.3205	2596442	1220712
PP-PSO	3215.168	440.3601	2543.372	324.3884	**1005761**	1123643
SopPSO	2646.873	257.1059	2284.727	145.7952	1157983	788348.7
	C19		C20		C21	
GCLPSO	**309824.8**	484215.9	2582.642	164.4844	2388.455	113.7288
FST-PSO	516084.8	698721.5	2899.979	231.6999	7244.879	2052.298
PP-PSO	41786.26	79193.27	2890.96	171.3033	**2256.39**	28.80583
SopPSO	54267.15	29390.6	**2516.419**	176.5408	2257.635	37.98601
	C22		C23		C24	
GCLPSO	**2334.505**	46.10819	**2998.283**	83.77263	3033.754	198.0154
FST-PSO	2525.269	45.64213	5159.713	754.2833	3490.721	199.574
PP-PSO	2423.022	33.20083	5430.172	566.2067	**2624.505**	7.079901
SopPSO	2417.927	37.82041	3145.016	106.5341	2905.707	359.1231
	C25		C26		C27	
GCLPSO	3232.716	227.5693	5730.447	572.2469	**3200.007**	0.000251
FST-PSO	4701.112	526.0779	7414.885	590.2477	5823.378	652.2105
PP-PSO	**3043.248**	83.18019	**3271.193**	1400.637	6554.57	553.1435
SopPSO	3065.398	36.85945	6913.551	1436.454	4305.518	328.8754
	C28		C29		C30	
GCLPSO	3330.014	140.045	**3545.836**	260.3361	**166532.5**	499410.3
FST-PSO	5767.055	589.9138	4628.107	484.319	1.61E + 08	1.53E + 08
PP-PSO	**3303.625**	44.57472	4378.405	348.2795	1822939	1242467
SopPSO	3481.639	401.8921	3955.652	199.209	2579572	2027674
	C31		C32		C33	
GCLPSO	**359.1043**	1029.417	**344.3299**	472.5213	**3.214955**	1.982597
FST-PSO	8846219	9085310	3228.755	1238.997	9.540943	0.826491
PP-PSO	2.35E + 08	1.38E + 08	26635.15	6358.773	6.644039	1.684984
SopPSO	1.54E + 08	1.09E + 08	9416.839	3062.053	8.462442	1.53715
	C34		C35		C36	
GCLPSO	**12.61567**	5.36528	**1.206667**	0.049533	**2.533985**	1.523304
FST-PSO	58.25209	22.16737	20.03394	12.11788	9.771933	1.106773

Table 5: Continued.

	Avg	Std	Avg	Std	Avg	Std
	C1		C2		C3	
PP-PSO	47.66576	8.638141	1.660454	0.233334	8.157233	1.265678
SopPSO	25.87162	8.484042	1.616367	0.114422	3.772674	1.242947
	C37		C38		C39	
GCLPSO	680.2373	258.1453	**3.326769**	0.477053	**1.134663**	4508.552
FST-PSO	1336.01	232.5401	4.777392	0.375858	1.601969	0.395181
PP-PSO	1119.353	385.7451	4.397828	0.271175	1.527999	0.23405
SopPSO	**648.257**	242.6093	4.063498	0.431445	1.257091	0.102787
	C40					
	Avg	Std	Rank	ARV	+/=/−	
GCLPSO	21.34823	0.080195	**1**	**1.11**		
FST-PSO	**21.14655**	0.084509	4	3.385	37/1/2	
PP-PSO	21.37964	0.086629	3	2.86	24/11/5	
SopPSO	21.22294	0.087032	2	2.3475	22/7/11	

C9, the adaptive values searched by the GCLPSO are superior to other peers. It is proved that the GCLPSO has better results on these multimodal functions than these improved PSO variants. At the same time, GCLPSO also has a better effect on hybrid functions and composition functions. On the benchmark functions of CEC2019, the adaptive values of the GCLPSO on the eight functions are better than other comparison algorithms. The Friedman test results show that the GCLPSO ranks first. As the results of "+/=/−"show, the GCLPSO is superior to the comparison algorithms in most functions.

In order to more intuitively and clearly understand the convergence trend of the algorithm and evaluate the performance of the algorithm, representative benchmark functions were selected from CEC2017 and CEC2019 for analysis. As shown in Figure 3, on CEC2017 multimodal benchmark functions C6 and C9, the adaptive values searched by the GCLPSO are better than other comparison algorithms. On the C6 and C9 benchmark functions, the adaptive value searched by the GCLPSO in the early iterations is not the best, but the adaptive value searched by the GCLPSO in later iterations is better than other algorithms. This is because this paper introduces the GWO algorithm idea into CLPSO to further improve its local search ability. On the C17 function, all the algorithms in the early iterations quickly searched for an adaptive value. In the later iterations, part of the comparison algorithms fell into a local optimum, and the GCLPSO continued to update the optimal fitness value. On the C18, C31, C33, and C38 benchmark functions, compared with other comparison algorithms, the GCLPSO not only searched for the best fitness value in the later iterations but also had a higher convergence trend and still updated the optimal fitness value. On the C23 and C27 benchmark functions, the GCLPSO converged to an adaptive value in the early stage of the iteration, and the trend of updating the adaptive value in the later iteration was relatively low. However, the adaptive value of the GCLPSO is much better than other comparison algorithms.

4.4. Feature Selection. Feature selection is a multiobjective optimization problem. The goal of this method is to select features as few as possible in the multifeature problem to obtain the greatest classification accuracy. In this section, the proposed GCLPSO was compared with the advanced feature selection algorithms on 12 different UCI datasets [86] as described in Table 6.

GCLPSO constantly updates the position of particles to achieve new solutions. Nevertheless, we need to select the feature of the problem in the binary form for feature selection problems. Therefore, we need to convert the continued values obtained from the algorithm into binary values. We first use the random threshold to reinitialize the algorithm to generate a binary value, as shown in the following equation:

$$x_{ij} = \begin{cases} 0, & \text{rand} \le 0.5, \\ 1, & \text{rand} > 0.5, \end{cases} \tag{13}$$

where x denotes the specific position value of the individual and i and j denote the i-th row and the j-th column, respectively.

Then, the continuous solution obtained in the algorithm is compressed by using the V-shaped transfer function to implement the conversion, and the function enables the search agent to move in the 0 to 1 space as shown in the following equation:

$$s = |\tanh(x)|, \tag{14}$$

where x is a continuous value.

The value achieved after conversion by the V-shaped transfer function is a continued value between 0 and 1. Finally, the binary value acquired by the initialization and the value achieved by the V-shaped transfer function are used to generate a new binary value by the following equation:

$$x = \begin{cases} \text{posOut} = \sim\text{pos}, & \text{rand} < s, \\ \text{posOut} = \text{pos}, & \text{rand} \ge s, \end{cases} \tag{15}$$

where *posOut* represents the newly achieved binary value, *pos* represents the initialized binary value, and s represents the continued value obtained by the V-shaped transfer function.

As with K-fold cross-validation, the data are separated into a training set and a test set. The verification data are

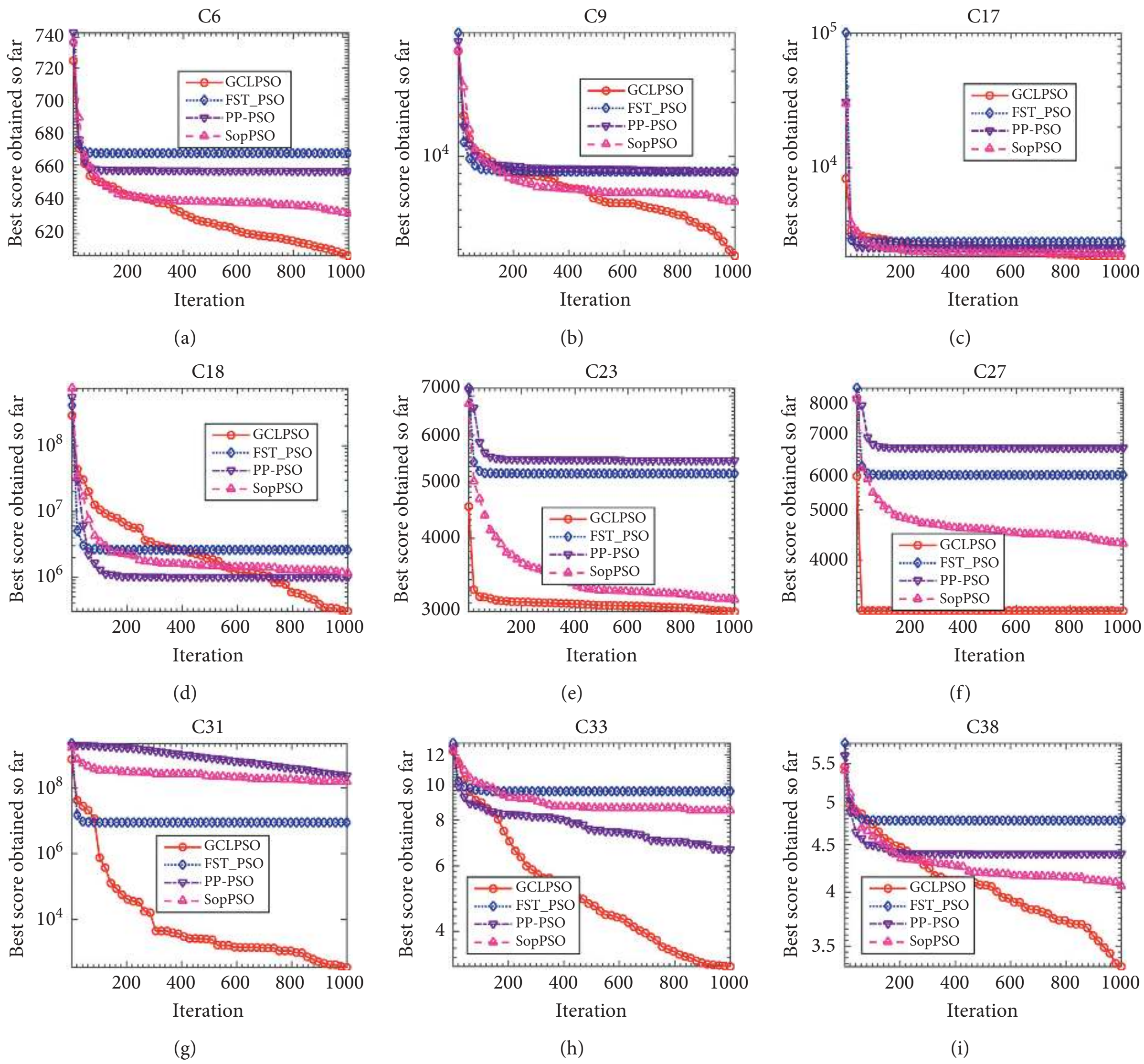

Figure 3: Convergence trends of the GCLPSO and other methods.

Table 6: Description of 12 datasets.

No.	Name	No. of features	No. of samples
D1	Breastcancer	10	699
D2	Exactly	14	1000
D3	Heart	14	270
D4	IonosphereEW	35	351
D5	Lymphography	19	148
D6	Vote	17	300
D7	WineEW	14	178
D8	Zoo	17	101
D9	wdbc	31	569
D10	Wielaw	31	240
D11	Australian	15	690
D12	Cleveland_heart	14	303

taken from the training data. The training set data are separated into K groups (K-fold), and each subset of the data is separately verified. One subset was used as the test data, and the remaining $K-1$ subset data were used as a training set. In the experiment, each dataset was run N times, each performing a K-fold cross-validation procedure. This experiment used the K-nearest neighbor (KNN) [87] classifier to classify the data. The KNN classifier uses the data in the training set to train the model and then uses the model to test the data in the validation set. Finally, the model is used to classify the test dataset to obtain the final accuracy. In this paper, the GCLPSO is binary-converted and compared with other feature selection methods including BGWO [86], BMFO [88], BPSO [89], and BBA [90] to estimate the performance of the proposed algorithm.

In this study, feature information is obtained in the binary form. Each of these search agents is a one-dimensional binary vector whose length relies on the number of features in the dataset. Each search agent represents a solution where an element value of 0 in the vector indicates that the feature is not chosen, and a value of 1 implied that the feature has been chosen. Then, each solution is evaluated by the fitness value acquired. The relationship between the error rate and the fitness value obtained by the classifier through data set validation is as shown in the following equation:

$$\text{fitness} = \alpha \times (1 - \text{correctRate}) + \beta \times \frac{N_i}{N}, \quad (16)$$

where *correctRate* represents the accuracy of the KNN classifier on the validation set, N represents the total number of features in the dataset, and N_i represents the number of features obtained by the i-th search agent after feature selection. In addition, α and β are two weights, respectively, where $\alpha = 0.95$ and $\beta = 0.05$. The entire process updates each search agent by iteration until the maximum iterations.

Table 7 illustrates the average fitness values acquired by the GCLPSO and BMFO, BGWO, BPSO, and BBA in 12 datasets and uses bold fonts to represent the best results. It can be observed that the GCLPSO possesses the best fitness value on the D2 to D5 datasets and the D9 and D11 datasets. In these datasets, the number of features in the D4 and D9 datasets is greater than 30. The average fitness value of the GCLPSO is superior to the BMFO algorithm on all datasets. Except that the average fitness value of the GCLPSO is inferior than the BPSO on the D6 dataset, the GCLPSO is greater than the BPSO on other datasets. In general, the average fitness of the GCLPSO algorithm on the 12 datasets is the best, which also shows that the algorithm works best in this kind of problem.

Table 8 demonstrates the average error rate of each algorithm on the verification set when K-fold cross-validation is performed for 12 datasets. As can be observed from the table, the GCLPSO has the lowest average error rate on the D4, D5, D6, and D9 datasets. At the same time, on the D2, D7, and D8 datasets, the average error rate of the GCLPSO algorithm is 0, which proves that the classification accuracy of the algorithm is 100% on these datasets. On the 12 datasets, BGWO has the lowest average error rate, and GCLPSO ranks second in the average error rate. This is because the excellence of the feature selection result is determined by two factors, namely, the selected feature number and the error rate. Although the average error rate of the algorithm in the 12 datasets is higher than that of the BGWO, the average fitness value of the 12 datasets is superior to the BGWO.

Table 9 shows the average number of features selected for each dataset after feature selection. In the D1, D4, and D9 datasets, the GCLPSO selects the fewest number of features. The GCLPSO averages the least number of selected features in 12 datasets.

The GCLPSO was tested on twelve datasets and compared with other feature selection methods to verify the effectiveness of the algorithm. Although the error rate of the GCLPSO is higher than that of the BGWO on 12 datasets, the optimal fitness value is obtained from the two factors that affect the feature selection effect, which proves the capability of the algorithm proposed.

4.5. Practical Constraint Modeling Problems. In this part, GCLPSO was used to solve 3 mathematical models with a constrained problem: I-beam, welded beam, and pressure vessel design problems. We need to choose an appropriate constraint processing method to solve mathematical problems with constraints. Coello Coello [91] described several kinds of penalty functions in detail. The death penalty is the most appropriate type of penalty functions. It constructs the primary target value of the mathematical model to be processed and uses a heuristic algorithm to eliminate the infeasible solution automatically. Therefore, GCLPSO combined with penalty functions was used in this experiment to solve the three famous mathematical model problems.

4.5.1. Welded Beam Design Problem. The purpose of the welding beam construction [92] problem is to obtain the minimum manufacturing cost of the model. Among them, four factors are influencing the manufacturing cost constraints, including the bucking load (P_c), the deflection rate (δ), the bending stress in the beam (θ), and the shear stress (τ). We control the manufacturing cost of the model through four optimization parameters, including the height of the bar (t), weld thickness (h), bar thickness (b), and bar length (l). The mathematical model description of the welding beam design problem is shown as follows:

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

Objective:

$f(\vec{x})_{\min} = 1.10471 x_2 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2)$.

Subject to

$$\begin{aligned}
g_1(\vec{x}) &= \tau(\vec{x}) - \tau_{\max} \le 0,\\
g_2(\vec{x}) &= \sigma(\vec{x}) - \sigma_{\max} \le 0,\\
g_3(\vec{x}) &= \delta(\vec{x}) - \delta_{\max} \le 0,\\
g_4(\vec{x}) &= x_1 - x_4 \le 0,\\
g_5(\vec{x}) &= P - P_C(\vec{x}) \le 0,\\
g_6(\vec{x}) &= 0.125 - x_1 \le 0,\\
g_7(\vec{x}) &= 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0.
\end{aligned} \quad (17)$$

Table 7: Average fitness value of the GCLPSO and other feature selection algorithms on the dataset.

	BMFO	BGWO	BPSO	BBA	GCLPSO
D1	$3.25E-02$	$3.22E-02$	$3.09E-02$	**2.96E − 02**	$2.99E-02$
D2	$4.21E-02$	$2.31E-02$	$2.31E-02$	$1.20E-01$	**2.31E − 02**
D3	$9.16E-02$	$7.83E-02$	$8.15E-02$	$7.90E-02$	**7.63E − 02**
D4	$1.19E-02$	$1.39E-02$	$1.87E-02$	$1.16E-02$	**8.81E − 03**
D5	$2.87E-02$	$2.57E-02$	$2.67E-02$	$2.95E-02$	**2.18E − 02**
D6	$2.50E-02$	$2.07E-02$	**1.93E − 02**	$2.15E-02$	$1.94E-02$
D7	$1.22E-02$	$1.19E-02$	$1.15E-02$	**1.02E − 02**	$1.04E-02$
D8	$1.04E-02$	$1.01E-02$	$1.04E-02$	**9.63E − 03**	$9.72E-03$
D9	$1.21E-02$	$1.20E-02$	$1.46E-02$	$1.01E-02$	**8.94E − 03**
D10	$6.17E-02$	**5.16E − 02**	$6.33E-02$	$5.81E-02$	$5.24E-02$
D11	$1.03E-01$	$9.86E-02$	$9.90E-02$	$1.06E-01$	**9.83E − 02**
D12	$9.43E-02$	**8.70E − 02**	$8.80E-02$	$9.14E-02$	$8.72E-02$
Average	4.3333	2.6667	3.3333	3	**1.5**
Rank	5	2	4	3	**1**

Table 8: Average error rate of the GCLPSO and other feature selection algorithms on datasets.

	BMFO	BGWO	BPSO	BBA	GCLPSO
D1	$1.36E-02$	$1.36E-02$	$1.10E-02$	**1.10E − 02**	$1.15E-02$
D2	$2.00E-02$	**0.00E + 00**	**0.00E + 00**	$9.71E-02$	**0.00E + 00**
D3	$7.52E-02$	**5.78E − 02**	$6.30E-02$	$6.26E-02$	$5.89E-02$
D4	$4.55E-03$	$3.38E-03$	$7.13E-03$	$3.94E-03$	**1.69E − 03**
D5	$1.84E-02$	$1.28E-02$	$1.46E-02$	$1.86E-02$	**1.06E − 02**
D6	$1.60E-02$	$9.99E-03$	$9.35E-03$	$1.30E-02$	**1.06E − 02**
D7	$5.56E-04$	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
D8	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
D9	$4.40E-03$	$2.63E-03$	$3.51E-03$	$2.45E-03$	**2.29E − 03**
D10	$5.58E-02$	**4.21E − 02**	$5.33E-02$	$5.04E-02$	$4.59E-02$
D11	$9.20E-02$	**8.12E − 02**	$8.13E-02$	$9.63E-02$	$8.60E-02$
D12	$7.89E-02$	**6.80E − 02**	$6.96E-02$	$7.79E-02$	$7.18E-02$
Average	4.25	**1.75**	2.5	3.0833	1.8333
Rank	5	**1**	3	4	2

Table 9: Average feature length of the GCLPSO and other feature selection algorithms on the dataset.

	BMFO	BGWO	BPSO	BBA	GCLPSO
D1	3.52	3.47	3.67	3.45	**3.43**
D2	**5.88**	6.01	6	6.06	6
D3	5.25	6.09	5.64	**5.08**	5.3
D4	5.18	7.27	8.11	5.37	**4.9**
D5	**4.07**	4.88	4.63	4.29	4.22
D6	3.13	3.57	3.34	**2.94**	2.98
D7	3.04	3.09	2.99	**2.66**	2.7
D8	3.33	3.23	3.33	**3.08**	3.11
D9	4.73	5.7	6.73	4.68	**4.06**
D10	**5.26**	6.99	7.59	6.1	5.27
D11	4.39	6.02	6.1	**4.09**	4.63
D12	5.02	5.83	5.69	**4.51**	4.92
Average	2.5	4.25	4.1667	2	**1.9167**
Rank	3	5	4	2	**1**

Variable ranges:

$$\begin{aligned} &0.1 \le x_1 \le 2,\\ &0.1 \le x_2 \le 10,\\ &0.1 \le x_3 \le 10,\\ &0.1 \le x_4 \le 2, \end{aligned} \tag{18}$$

where

$$\begin{aligned} \tau(\vec{x}) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\\ \tau' &= \frac{P}{\sqrt{2}x_1x_2},\\ \tau'' &= \frac{MR}{J},\\ M &= P\left(L + \frac{x_2}{2}\right),\\ R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\\ J &= 2\left\{\sqrt{2}\,x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\ \sigma(\vec{x}) &= \frac{6PL}{x_4x_3^2},\\ \delta(\vec{x}) &= \frac{6PL^3}{Ex_3^2x_4},\\ P_C(\vec{x}) &= \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\ P &= 60001b,\\ L &= 14\text{in},\\ \delta_{\max} &= 0.25\text{ in},\ldots,\\ E &= 30 \times 1^6\text{ psi},\\ G &= 12 \times 10^6\text{ psi},\\ \tau_{\max} &= 13600\text{ psi},\\ \sigma_{\max} &= 30000\text{ psi}. \end{aligned} \tag{19}$$

This engineering design model has attracted the attention of many researchers. Kaveh and Khayatazad used RO [93] to process this mathematical model. Lee and Geem [94] used HS to optimize the mathematical model. As shown in Table 10, the model was optimized by the HS method, and the optimal manufacturing cost was 2.3807. The improved HS (IHS) [95] also optimized the model, and the optimal manufacturing cost was 1.7248. Radgsdell and Phillips [96] used mathematical methods such as Davidon–Fletcher–Powell and simplex method to solve the optimal cost.

The mathematical model is optimized with the solution of the GCLPSO, and the results are compared with those of other methods. As shown in Table 10, the minimum manufacturing cost optimized by the GCLPSO is 1.715355, indicating that when the four parameter values are set to 0.20799, 3.25802, 9.02820, and 0.208064, respectively, the manufacturing cost of this model can reach 1.715355, and the minimum cost is less than the results obtained by other methods. It is shown that the proposed GCLPSO algorithm can optimize the mathematical model and obtain the minimum design and manufacturing cost of the welded beam.

4.5.2. Pressure Vessel Design Problem. The purpose of this mathematical model is to minimize the cost of cylindrical pressure vessels [97]. Among them, the manufacturing cost of the model is closely associated with welding, material, and structure. The end of the model is covered, and the head part is a hemispherical figure. We reduce the manufacturing cost of the model by optimizing the variable internal radius (R), head thickness (T_h), shell thickness (T_s), and cross-section range minus head (L). The mathematical expression of the model is shown as follows:

Consider $\vec{x} = [\,x_1\;\; x_2\;\; x_3\;\; x_4\,] = [\,T_s\;\; T_h\;\; R\;\; L\,]$.

Objective: $f(\vec{x})_{\min} = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$.

Subject to

$$\begin{aligned} g_1(\vec{x}) &= -x_1 + 0.0193x_3 \le 0,\\ g_2(\vec{x}) &= -x_3 + 0.00954x_3 \le 0,\\ g_3(\vec{x}) &= -\pi x_4x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0,\\ g_4(\vec{x}) &= x_4 - 240 \le 0. \end{aligned} \tag{20}$$

Variable ranges:

$$\begin{aligned} &0 \le x_1 \le 99,\\ &0 \le x_2 \le 99,\\ &10 \le x_3 \le 200,\\ &10 \le x_4 \le 200. \end{aligned} \tag{21}$$

This mathematical model has been tried by many researchers to find the optimal value. As shown in Table 11, He and Wang [98] used a particle swarm optimization algorithm to optimize the model, and the optimized manufacturing cost was 6061.0777. Deb [99] optimized the model by the genetic algorithm, and the manufacturing cost

Table 10: Comparison of the GCLPSO with other methods for the welding beam design problem.

Technique	Best variables				Best cost
	H	l	t	b	
GCLPSO	0.20799	3.25802	9.02820	0.208064	1.715355
WOA [24]	0.205396	3.484293	9.037426	0.206276	1.730499
RO [93]	0.203687	3.528467	9.004233	0.207241	1.735344
HS [94]	0.2442	6.2231	8.2915	0.2433	2.3807
IHS [95]	0.20573	3.47049	9.03662	0.20573	1.7248
Random [96]	0.4575	4.7313	5.0853	0.6600	4.1185
Simple [96]	0.2792	5.6256	7.7512	0.2796	2.5307
David [96]	0.2434	6.2552	8.2915	0.2444	2.3841

Table 11: Comparison of the GCLPSO with other methods for the pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
GCLPSO	0.784508	0.387656	40.6289	195.8892	5989.654
IHS [95]	1.125000	0.625000	58.29015	43.69268	7197.7300
PSO [98]	0.812500	0.437500	42.091266	176.746500	6061.0777
GA [99]	0.937500	0.500000	48.329000	112.679000	6410.3811
ES [100]	0.812500	0.437500	42.098087	176.640518	6059.7456
Lagrangian multiplier [97]	1.125000	0.625000	58.291000	43.690000	7198.0428
Branch and bound [101]	1.125000	0.625000	47.700000	117.71000	8129.1036

Table 12: Comparison of the GCLPSO with other methods for the I-beam design problem.

Algorithm	Best variables				Optimum vertical
	b	h	t_w	t_f	
GCLPSO	50	80	1.764669	5	0.00662596
MFO [29]	50	80	1.7647	5	0.0066259
ARSM [102]	48.42	79.99	0.90	2.40	0.0157
IARSM [102]	0.2442	6.2231	8.2915	0.2433	0.131
CS [103]	50	80	0.9	2.321675	0.0130747
SOS [104]	50	80	0.9	2.32179	0.0130741

was 6410.3811. Also, some scholars used IHS [95], ES [100], and mathematical methods [97, 101] to solve the optimal solution.

The optimal value obtained by optimizing the mathematical model through the GCLPSO is 5989.654, indicating that when the parameter values of T_s, T_h, R, and L are set to 0.784508, 0.387656, 40.6289, and 195.8892, respectively, the total cost of cylindrical pressure vessels is the minimum. GCLPSO algorithm and other methods utilized to find the optimal design scheme are shown in Table 11. It can be seen that the GCLPSO algorithm provides a better solution for this model.

4.5.3. I-Beam Design Problem. The mathematical model aims to optimize the structure of the I-beam. The design of the I-beam is improved to achieve the minimum vertical deflection. The optimal parameters of the model are the length, height, and two thicknesses. The mathematical formula of this model can be described as follows:

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

Objective: $f(\vec{x})_{\min} = (5000/(t_w(h-2t_f)^3/12) + (bt_f^3/6) + 2bt_f(h-t_f/2)^2)$.

Subject to

$$g(\vec{x}) = 2bt_w + t_w\left(h - 2t_f\right) \leq 0. \tag{22}$$

Variable ranges:

$$\begin{aligned} &10 \leq x_1 \leq 50,\\ &10 \leq x_2 \leq 80,\\ &0.9 \leq x_3 \leq 5,\\ &0.9 \leq x_4 \leq 5. \end{aligned} \tag{23}$$

Some scholars use different methods to optimize the model. Wang used ARSM [102] to optimize the model to obtain the minimum vertical deflection and also used the improved IARSM [102] to optimize the model. Gandomi et al. used CS [103] to solve the minimum vertical deflection of the model. Cheng and Prayogo used SOS [104] to optimize this problem.

As shown in Table 12, the optimization results of the GCLPSO algorithm were compared with those of other methods before, and the minimum vertical deflection obtained by the GCLPSO was 0.00662596. This observation indicates that when the parameters are set to 50, 80, 1.764669, and 5, respectively, the vertical deflection of the I-beam is 0.00662596. As can be seen from the table, GCLPSO can also provide a good solution for this model.

5. Conclusion and Future Directions

This paper presents an improved algorithm named GCLPSO. This algorithm introduces the GWO into CLPSO to improve the local search capability of the CLPSO. The GCLPSO achieves a more stable status between global search and local search, which boosts the ability to search for the optimal solution. The improved algorithm was compared with seven classical MAs and eight advanced metaheuristic algorithms on the CEC2017 benchmark functions. Experiments were carried out in the same experimental environment, and the experimental results have shown that the algorithm proposed had distinct advantages over other comparison algorithms in terms of CEC2017 benchmark functions with four different types. It was proved that GCLPSO has the strong searching ability on the benchmark functions. In this paper, the improved algorithm was binary-transformed and compared with other algorithms when coping with feature selection on 12 datasets, which proves that the algorithm has a good effect on feature selection. Moreover, the algorithm was also applied to three practical engineering design problems: pressure vessel, I-beam, and welded beam design problems. The results of algorithm optimization were compared with the results obtained by other methods. GCLPSO has achieved good optimization results in all three engineering problems. It shows that this algorithm can deal with the constraint problem effectively at the same time.

In the future research work, the algorithm can be used in many aspects. For instance, the algorithm can be used to optimize the machine learning models such as neural networks [105–110]. On the contrary, the performance of the algorithm can be improved further and can also be extended to multiobjective directions [111]. It is also possible to explore a parallel computing framework based on the algorithm of this paper, which is applied to more complex optimization problems. At the same time, it can also be combined with finance, agriculture, and other applications [109, 112, 113] to explore the best practical application scenarios of the algorithm. Therefore, there are still many aspects for us to explore and discover as a further study to be implemented.

Data Availability

The data involved in this study are all public data, which can be downloaded through public channels.

Conflicts of Interest

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

Authors' Contributions

Xianchang Wang, Helong Yu, and Huiling Chen contributed equally to this work.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (U1809209), Science and Technology Development Project of Jilin Province (20190301024NY and 20190302117GX), and Jilin Provincial Industrial Innovation Special Fund Project (2018C039-3). Thanks to the efforts of Ali Asghar Heidari during the preparation of this research.

References

[1] M. Mafarja, I. Aljarah, A. A. Heidari et al., "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowledge-Based Systems*, vol. 161, pp. 185–204, 2018.

[2] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[3] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, and H. Chen, "Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton," *Applied Soft Computing*, vol. 24, pp. 585–596, 2014.

[4] L. Shen, H. Chen, Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[5] M. Wang, H. Chen, B. Yang et al., "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, 2017.

[6] A. G. Hussien, M. Wang, and G. Liang, "Crow search algorithm: theory, recent advances, and applications," *IEEE Access*, 2020.

[7] H. Zhang, Z. Cai, X. Ye et al., "A multi-strategy enhanced salp swarm algorithm for global optimization," *Engineering with Computers*, 2020.

[8] X. Liang, Z. Cai, M. Wang, X. Zhao, H. Chen, and C. Li, "Chaotic oppositional sine-cosine method for solving global optimization problems," *Engineering with Computers*, 2020.

[9] A. F. Ba, H. Huang, M. Wang et al., "Levy-based antlion-inspired optimizers with orthogonal learning scheme," *Engineering with Computers*, 2020.

[10] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," *Engineering with Computers*, 2020.

[11] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Applied Soft Computing*, vol. 86, Article ID 105884, 2019.

[12] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, and C. Li, "Chaos enhanced bacterial foraging optimization for global optimization," *IEEE Access*, vol. 6, pp. 64905–64919, 2018.

[13] X. Xu and H.-l. Chen, "Adaptive computational chemotaxis based on field in bacterial foraging optimization," *Soft Computing*, vol. 18, no. 4, pp. 797–807, 2014.

[14] H. Chen, S. Jiao, M. Wang, A. A. Heidari, and X. Zhao, "Parameters identification of photovoltaic cells and modules

using diversification-enriched Harris hawks optimization with chaotic drifts," *Journal of Cleaner Production*, vol. 244, Article ID 118778, 2019.

[15] Z. Cai, J. Gu, J. Luo et al., "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Systems with Applications*, vol. 138, Article ID 112814, 2019.

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

[17] X. Zhao, X. Zhang, Z. Cai et al., "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Computational Biology and Chemistry*, vol. 78, pp. 481–490, 2019.

[18] H. Chen, A. A. Heidari, H. Chen, M. Wang, Z. Pan, and A. H. Gandomi, "Multi-population differential evolution-assisted Harris hawks optimization: framework and case studies," *Future Generation Computer Systems*, vol. 111, pp. 175–198, 2020.

[19] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[20] H. Faris, A. M. Al-Zoubi, A. A. Heidari et al., "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion*, vol. 48, pp. 67–83, 2019.

[21] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[22] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol. 7, pp. 20281–20292, 2019.

[23] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[24] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[25] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Applied Mathematical Modelling*, vol. 71, pp. 45–59, 2019.

[26] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Applied Mathematical Modelling*, vol. 73, pp. 109–123, 2019.

[27] M. Wang and H. Chen, "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis," *Applied Soft Computing*, vol. 88, Article ID 105946, 2019.

[28] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, "An efficient double adaptive random spare reinforced whale optimization algorithm," *Expert Systems with Applications*, vol. 154, Article ID 113018, 2019.

[29] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[30] Y. Xu, H. Chen, A. A. Heidari et al., "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Systems with Applications*, vol. 129, pp. 135–155, 2019.

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

[32] J. Luo, H. Chen, Q. Zhang, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Applied Mathematical Modelling*, vol. 64, pp. 654–668, 2018.

[33] M. Mafarja, I. Aljarah, A. A. Heidari et al., "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Systems*, vol. 145, pp. 25–45, 2018.

[34] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," *Soft Computing*, vol. 23, no. 17, pp. 7941–7958, 2018.

[35] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Studies in Computational Intelligence*, pp. 65–74, Springer, Berlin, Germany, 2010.

[36] H. Yu, N. Zhao, P. Wang, H. Chen, and C. Li, "Chaos-enhanced synchronized bat optimizer," *Applied Mathematical Modelling*, vol. 77, pp. 1201–1215, 2020.

[37] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[38] 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.

[39] 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.

[40] H. Chen, S. Li, A. Asghar Heidari et al., "Efficient multi-population outpost fruit fly-driven optimizers: framework and advances in support vector machines," *Expert Systems with Applications*, vol. 142, Article ID 112999, 2020.

[41] X. Zhang, Y. Xu, C. Yu et al., "Gaussian mutational chaotic fruit fly-built optimization and feature selection," *Expert Systems with Applications*, vol. 141, Article ID 112976, 2020.

[42] H. Chen, M. Wang, and X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation*, vol. 369, Article ID 124872, 2020.

[43] H. Chen, S. Jiao, A. A. Heidari, M. Wang, X. Chen, and X. Zhao, "An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models," *Energy Conversion and Management*, vol. 195, pp. 927–942, 2019.

[44] H. Chen, A. A. Heidari, X. Zhao, L. Zhang, and H. Chen, "Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies," *Expert Systems with Applications*, vol. 144, Article ID 113113, 2020.

[45] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Western Australia, November 1995.

[46] H. Pan, L. Wang, and B. Liu, "Particle swarm optimization for function optimization in noisy environment," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 908–919, 2006.

[47] J. I. Ababneh and M. H. Bataineh, "Linear phase FIR filter design using particle swarm optimization and genetic algorithms," *Digital Signal Processing*, vol. 18, no. 4, pp. 657–668, 2008.

[48] S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," *Electric Power Systems Research*, vol. 72, no. 3, pp. 203–212, 2004.

[49] H. E. Mostafa, M. A. El-Sharkawy, A. A. Emary, and K. Yassin, "Design and allocation of power system stabilizers using the particle swarm optimization technique for an interconnected power system," *International Journal of Electrical Power & Energy Systems*, vol. 34, no. 1, pp. 57–65, 2012.

[50] L. Hu, F. Lin, H. Li et al., "An intelligent prognostic system for analyzing patients with paraquat poisoning using arterial blood gas indexes," *Journal of Pharmacological and Toxicological Methods*, vol. 84, pp. 78–85, 2017.

[51] H. l. Chen, B. Yang, S. j. Wang et al., "Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy," *Applied Mathematics and Computation*, vol. 239, pp. 180–197, 2014.

[52] J. Zhu, X. Zhao, H. Li, H. Chen, and G. Wu, "An effective machine learning approach for identifying the glyphosate poisoning status in rats using blood routine test," *IEEE Access*, vol. 6, pp. 15653–15662, 2018.

[53] W. Deng, H. Zhao, X. Yang, J. Xiong, M. Sun, and B. Li, "Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment," *Applied Soft Computing*, vol. 59, pp. 288–302, 2017.

[54] M. Hu, T. Wu, and J. D. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 705–720, 2013.

[55] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, Seoul, South Korea, May 2001.

[56] D. Chen and C. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing*, vol. 9, no. 1, pp. 39–48, 2009.

[57] S. T. Hsieh, T. Y. Sun, C. C. Liu, and S. J. Tsai, "Efficient population utilization strategy for particle swarm optimizer," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 444–456, 2009.

[58] Y.-J. Gong, J.-J. Li, Y. Zhou et al., "Genetic learning particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2016.

[59] Z. H. Zhan, J. Zhang, and O. Liu, "Orthogonal learning particle swarm optimization," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO*, Montreal, Canada, 2009.

[60] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2015.

[61] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[62] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[63] D. Sánchez, P. Melin, and O. Castillo, "A grey wolf optimizer for modular granular neural networks for human recognition," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 4180510, 26 pages, 2017.

[64] M. Wang, H. Chen, H. Li et al., "Grey wolf optimization evolving kernel extreme learning machine: application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 54–68, 2017.

[65] P. Yao, H. Wang, and H. Ji, "Multi-UAVs tracking target in urban environment by model predictive control and Improved Grey Wolf Optimizer," *Aerospace Science and Technology*, vol. 55, pp. 131–143, 2016.

[66] Q. Li, H. Chen, H. Huang et al., "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Computational and Mathematical Methods in Medicine*, vol. 2017, Article ID 9512741, 15 pages.

[67] L. Hu, H. Li, Z. Cai et al., "A new machine-learning method to prognosticate paraquat poisoned patients by combining coagulation, liver, and kidney indices," *PLoS One*, vol. 12, no. 10, Article ID e0186427, 2017.

[68] J. Zhu, F. Zhu, S. Huang et al., "A new evolutionary machine learning approach for identifying pyrene induced hepatotoxicity and renal dysfunction in rats," *IEEE Access*, vol. 7, pp. 15320–15329, 2018.

[69] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," *Expert Systems with Applications*, vol. 86, pp. 64–76, 2017.

[70] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Technical Report, Nanyang Technological University, Singapore, 2016.

[71] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[72] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133

[73] N. Kumar, I. Hussain, B. Singh, and B. K. Panigrahi, "Single sensor-based MPPT of partially shaded PV system for battery charging by using Cauchy and Gaussian sine cosine optimization," *IEEE Transactions on Energy Conversion*, vol. 32, no. 3, pp. 983–992, 2017.

[74] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019–1043, 2018.

[75] F. Ye, X. Y. Lou, and L. F. Sun, "An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications," *PLoS One*, vol. 312, no. 4, Article ID e0173516, 2017.

[76] W. Wang and X. Liu, "Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm," *Chemometrics and Intelligent Laboratory Systems*, vol. 141, pp. 79–87, 2015.

[77] Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE Access*, vol. 5, pp. 6168–6186, 2017.

[78] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in Arabic sentiment analysis," *Applied Intelligence*, vol. 49, no. 5, pp. 1688–1707, 2019.

[79] X. Chen, H. Tianfield, C. Mei, W. Du, and G. Liu, "Biogeography-based learning particle swarm optimization," *Soft Computing*, vol. 21, no. 24, pp. 7519–7541, 2017.

[80] 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.

[81] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[82] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.

[83] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy self-tuning PSO: a settings-free algorithm for global optimization," *Swarm & Evolutionary Computation*, vol. 39, pp. 70–85, 2017.

[84] H. Zhang, M. Yuan, Y. Liang, and Q. Liao, "A novel particle swarm optimization based on prey-predator relationship," *Applied Soft Computing*, vol. 68, pp. 202–218, 2018.

[85] X. Xia, B. Wang, C. Xie, Z. Hu, B. Wei, and C. Jin, "A sophisticated PSO based on multi-level adaptation and purposeful detection," *Soft Computing*, vol. 22, pp. 2603–2618, 2018.

[86] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[87] P. Hart, "The condensed nearest neighbor rule (Corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515-516, 1968.

[88] G. I. Sayed, A. Darwish, and A. E. Hassanien, "Binary whale optimization algorithm and binary moth flame optimization with clustering algorithms for clinical breast cancer diagnoses," *Journal of Classification*, vol. 37, pp. 66–96, 2019.

[89] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.

[90] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 663–681, 2014.

[91] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[92] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[93] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: ray optimization," *Computers & Structures*, vol. 112-113, pp. 283–294, 2012.

[94] 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. 36–38, pp. 3902–3933, 2005.

[95] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[96] K. M. Ragsdell and D. T. Phillips, "Optimal design of a class of welded structures using geometric programming," *Journal of Engineering for Industry*, vol. 98, no. 3, pp. 1021–1025, 1976.

[97] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, no. 2, pp. 405–411, 1994.

[98] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.

[99] K. Deb, "GeneAS: a robust optimal design technique for mechanical component design," in Evolutionary Algorithms in Engineering Applications,D. Dasgupta and Z. Michalewicz, Eds., Springer, Berlin, Germany, pp. 497–514, 1997.

[100] 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.

[101] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.

[102] 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.

[103] 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.

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

[105] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2020.

[106] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3840–3854, 2020.

[107] X. Zhang, T. Wang, J. Wang, G. Tang, and L. Zhao, "Pyramid channel-based feature attention network for image dehazing," *Computer Vision and Image Understanding*, vol. 197-198, Article ID 103003, 2020.

[108] Y. Li, W.-G. Cui, H. Huang, Y.-Z. Guo, K. Li, and T. Tan, "Epileptic seizure detection in EEG signals using sparse multiscale radial basis function networks and the Fisher vector approach," *Knowledge-Based Systems*, vol. 164, pp. 96–106, 2019.

[109] Y. Liu, G. Chong, A. A. Heidari et al., "Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models," *Energy Conversion and Management*, vol. 223, Article ID 113211, 2020.

[110] Y. Li, Y. Liu, W.-G. Cui, Y.-Z. Guo, H. Huang, and Z.-Y. Hu, "Epileptic seizure detection in EEG signals using a unified temporal-spectral squeeze-and-excitation network," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 4, pp. 782–794, 2020.

[111] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization

problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2019.

[112] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, "Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3343–3355, 2019.

[113] X. Zhang, M. Fan, D. Wang, P. Zhou, and D. Tao, "Top-*k* feature selection framework using robust 0-1 integer programming," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.