IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2021, accepted November 30, 2021, date of publication December 3, 2021, date of current version December 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132617

A Novel Hybrid Hunger Games Search Algorithm With Differential Evolution for Improving the Behaviors of Non-Cooperative Animals

SHAOLANG LI[1], XIAOBO LI[1,2], HUI CHEN[1], YUXIN ZHAO[1], AND JUNWEI DONG[1]
[1]College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China
[2]College of Engineering, Lishui University, Lishui 323000, China

Corresponding author: Xiaobo Li (lxb@zjnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61373057, and in part by the Science and Technology Planning Project of Lishui City under Grant 2019RC05.

ABSTRACT Inspired by the behaviors of animals in the state of starvation, hunger games search algorithm (HGS) is proposed. HGS has shown competitive performance among other meta-heuristic (MH) algorithms. However, HGS tends to stagnate in local optimal for some complex optimization problems and remains premature convergence. Therefore, to solve these problems and enhance the diversity of the population, a modified HGS based on the operators of the differential evolution algorithm (DE), chaotic local search (CLS) strategies, and evolutionary population dynamics technique (EPD) is proposed (named DECEHGS). The proposed DECEHGS algorithm consists of two stages: in the first stage, based on the animals' behaviors, we use different evolutionary methods to update animals' positions; in the second stage, the CLS strategy and EPD technique are combined to prevent premature convergence and stagnation in a local optimum. The proposed algorithm was evaluated using IEEE CEC2014 and IEEE CEC2017 mathematical functions and four engineering problems. The experimental results demonstrate that DECEHGS has competitive performance in global optimization tasks and engineering problems compared with state-of-the-art algorithms.

INDEX TERMS Hunger games search, differential evolution algorithm, chaotic local search, evolutionary population dynamics, optimization.

I. INTRODUCTION

Many meta-heuristic (MH) algorithms have been proposed to solve real-world problems and obtain better performances, such as neural networks [1], [2], image thresholding [3], task scheduling on cloud computing [4], [5], ice manufacturing industry [6], feature selection [7], [8], data clustering problems [9] and distributed cloud framework [10]. Many phenomena in nature are a source of inspiration for algorithms. The diversity of phenomena leads researchers to propose several optimization approaches derived from swarm intelligence, physical principles, and coevolution between organisms.

MH algorithms can be divided into three categories based on different inspirations as evolutionary computation (EC) [11], swarm intelligence algorithms (SIA), and physical phenomena algorithms (PPA).

The first category EC is inspired by Darwin's principles of nature's capability to evolve the individuals to be well adapted to their environment. The most popular algorithms include genetic algorithm (GA) [12], differential evolution algorithm (DE) [13], biogeography-based optimization (BBO) [14] and evolution strategies (ES) [15].

The second category SIA is inspired by the behaviors of the swarm and their environments, such as harmony search algorithm (HS) [16], nomadic people optimizer (NPO) [17], whale optimizer algorithm (WOA) [18], particle swarm optimization (PSO) [19], grey wolf optimization (GWO) [20], harris hawk algorithm (HHO) [21] and mine blast algorithm (MBA) [22].

The third category PPA simulates the physical phenomena, such as multi-verse optimizer (MVO) [23], gravitational search algorithm (GSA) [24], symbiotic organisms search

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

(SOS) [25], charged system search (CSS) [26], chemical reaction optimization (CRO) [27] and water wave optimization (WWO) [28].

Inspired by these ideologies, many improved algorithms have been proposed, such as hybrid harmony search algorithm with grey wolf optimizer (GWO-HS) [29], coevolutionary particle swarm optimization with bottleneck objective learning strategy (CPSO) [30], fuzzy gravitational search algorithm (FGSA) [31] and differential evolution algorithm with strategy adaptation and knowledge-based control parameters (SAKPDE) [32].

Hungry games search algorithm (HGS) [33] is a new and efficient meta-heuristic algorithm inspired by the behavior of animals in nature when they are hungry. Like other MH methods, HGS consists of two phases (i.e., cooperative and non-cooperative behaviors), and each phase is determined according to a given value. HGS was reported to be superior to other recognized optimizers, such as PSO, GA, and gravitational search algorithm (GSA) [34], etc. However, HGS will quickly converge on the local optimum when solving complex problems, which is a widely existing problem of the MH algorithms [35], [36]. Another problem is the stagnation in a local optimum. MH methods' success in solving this problem depends on balancing the exploration and exploitation or hybridizing other algorithms. The original operators of non-cooperative animals in HGS fail to jump out of the local optimal, which will significantly degrade the quality of the solutions.

The DE algorithm was first proposed in 1994 by Storn and Price. DE is a simple and powerful evolutionary algorithm. The inherent simplicity of DE has drawn the attention of many researchers. For example, in [37], Pant *et al.* proposed a hybrid version of DE with PSO, and the results show that the proposed DE-PSO is reasonably competent for solving the benchmark functions and real-world problems. Wu *et al.* [38] presented a multi-population-based ensemble DE (MPEDE), in which the whole population is divided into three equally sized subpopulations and one larger subpopulation. Three different mutation strategies named "current-to-pbest/1", "current-to-rand/1", and "rand/1" are applied to the three populations, the controlling parameters of each strategy are made adaptively. Zou *et al.* [39] proposed a modified DE named MDE, employing Gaussian and uniform distribution to adjust scale factor and crossover rate.

We proposed an improved version of the HGS algorithm (DECEHGS) using the DE algorithm to improve the non-cooperative animals of the classic HGS algorithm to overcome these limitations. Moreover, evolutionary population dynamics technique (EPD) [40] and chaotic local search (CLS) [41] strategy are used to update the worst solutions and best solutions during the optimization process, which will reduce the effect of worst solutions on the quality of the population and the possibility of local optimum stagnation.

The main contribution of the current study can be outlined as follows:

1. The operators of DE are used to enhance the non-cooperative animals.
2. Apply the EPD technique to update the worst solutions.
3. CLS strategy is employed to update the best solution to avoid falling into local optimum.
4. The performance of the DECEHGS is evaluated using CEC2014 and CEC2017 test suites and four engineering problems.

The remainder of this paper is organized as follows: Section 2 briefly introduces the basic version of the HGS algorithm. We introduce the DE algorithm, EPD technique, and CLS strategy in Section 3. In Section 4, the DECEHGS algorithm is proposed. The comparison results are presented and discussed in Section 5. Finally, conclusions and future work are summarized in Section 6.

II. HUNGER GAMES SEARCH ALGORITHM

The hunger games search algorithm is a new MH algorithm proposed by Yang [33], which is inspired by the activities and behaviors of animals in a state of starvation. In HGS, animals mainly have two behaviors, cooperation and non-cooperation. The formula of cooperative behavior is shown as follows:

$$X(t+1) = X(t) \cdot (1 + randn) \tag{1}$$

where $X(t)$ represents the position of the current individual, *randn* is a random number that satisfies a normal distribution.

The formula of non-cooperative behavior is shown as follows:

$$X(t+1) = \begin{cases} W_1 \cdot X_b + R \cdot W_2 \cdot |X_b - X(t)|, & r_2 > E \\ W_1 \cdot X_b - R \cdot W_2 \cdot |X_b - X(t)|, & r_2 < E \end{cases} \tag{2}$$

where X_b represents the best individual at the current iteration, W_1 and W_2 represent the weight of hungry, r_2 is a random number in the range of [0,1], R is a ranging controller, and E represents the variable number that controls the global location, and their formulas are shown as follows:

$$E = \text{sech}(|F(i) - BF|) \tag{3}$$

where $F(i)$ represents the fitness value of each individual, BF is the best fitness at the current iteration, and the formula of **sech** is shown as follows:

$$\text{sech}(x) = \frac{2}{e^x + e^{-x}} \tag{4}$$

The equations of parameters R, W_1, and W_2 are shown as follows:

$$R = 2 \times shrink \times rand - shrink \tag{5}$$

$$shrink = 2 \times \left(1 - \frac{t}{T}\right) \tag{6}$$

$$W_1(i) = \begin{cases} hungry(i) \cdot \dfrac{N}{SHungry} \times r_4, & r_3 < l \\ 1, & r_3 > l \end{cases} \tag{7}$$

$$W_2(i) = (1 - \exp(-|hungry(i) - SHungry|)) \times r_5 \times 2 \tag{8}$$

where *rand* is a random number in the range of [0, 1], t is the current iteration, T is the maximum number of iterations, *hungry* represents the hunger for each individual, N means the swarm size of the individuals, *SHungry* is the sum of hungry feelings of all individuals, and r_3, r_4, r_5 are random numbers in the range of [0, 1].

The formula of *hungry(i)* is calculated as below:

$$hungry(i) = \begin{cases} 0, & AllFitness(i) == BF \\ hungry(i) + H, & AllFitness! = BF \end{cases} \tag{9}$$

where *AllFitness(i)* preserves each individual's fitness in the current iteration, and the best individual's hunger value is set to 0 in the current iteration. For the other individuals, a new hunger value H is added based on the former hunger value. The corresponding parameter H of each individual is different.

The formula of H is shown as follows:

$$H = \begin{cases} LH \times (1 + r_6), & TH < LH \\ TH, & TH \geq LH \end{cases} \tag{10}$$

$$TH = \frac{F(i) - BF}{WF - BF} \times r_7 \times 2 \times (ub_i - lb_i) \tag{11}$$

where r_6 and r_7 represent a random number in the range of [0, 1], LH is a limited parameter, $F(i)$ represents the fitness value of each individual, BF and WF are the best fitness and worst fitness value obtained from the current iteration, respectively, ub_i and lb_i represent the upper and lower bounds of the search space.

III. PRELIMINARIES

The DE algorithm is applied to enhance the behaviors of non-cooperative animals. Moreover, it can be found that the worst solution in the population may influence the performance of the algorithm and prevent it from converging on the global optimum. Therefore, we integrate the updated solutions with the concept of CLS and EPD to avoid premature convergence and stagnation in the local optimum. The EPD technique replaces the worst agents by generating new agents in the neighborhood of the better ones. The CLS strategy creates a new agent randomly and chaotically in the neighborhood of the old agent.

A. DIFFERENTIAL EVOLUTION ALGORITHM

DE algorithm has been proved to be a simple and efficient algorithm. DE works in two phases: initialization and evolution. In the first phase, the population is generated randomly. The generated population mainly goes through three processes in the second phase: mutation, crossover, and selection. DE repeats the second process until the termination criteria are satisfied. During initialization, each individual in the current iteration is generated as follows:

$$X_i = lb_i + (ub_i - lb_i) \cdot rand \tag{12}$$

where $rand \in [0, 1]$, lb_i and ub_i are lower and upper bounds of search space, respectively.

In the mutation phase, a mutant vector V_i is generated for each individual as below:

$$V_i = X_{r_8} + F \cdot (X_{r_9} - X_{r_{10}}) \tag{13}$$

where F is the scaling factor, and the value of F is varied from 1 to 0, r_8, r_9, r_{10} are random integer numbers in the range of $[1, N]$.

After the mutation phase, the crossover phase generates a new vector called trial vector U_i, which is denoted as follows:

$$U_i = \begin{cases} V_i, & if\ rand \leq Cr \\ X_i, & otherwise \end{cases} \tag{14}$$

where $Cr \in [0,1]$.

In the selection phase, a comparison between the target and trial vector is made according to their fitness value. This operation performs as below:

$$X_i = \begin{cases} U_i, & if\ f(U_i) \leq f(X_i) \\ X_i, & otherwise \end{cases} \tag{15}$$

where f denotes the fitness function.

B. CHAOTIC LOCAL SEARCH

Chaos is a typical nonlinear phenomenon in nature, which is characterized by randomness, ergodicity, and sensitivity to initial conditions. Because of the randomness and ergodicity, chaos optimization works well in the small search space, but it requires unacceptable optimization time in the ample search space. Therefore, chaotic search is frequently incorporated into other global optimizers such as GA, PSO, and WOA to enhance their searching capacities.

The mechanism of CLS can not only make the algorithm more capable of avoiding falling into the local optimum but also enhance the searching capacities and make a better harmony between exploration and exploitation.

In [42] and [43], Wei and Mascia *et al.* used local search to construct the solutions to the objective function. However, standard local search may inevitably lead to rapid convergence or stagnation in local optima. Owing to its randomness and ergodicity, CLS can effectively overcome these shortcomings.

C. EVOLUTIONARY POPULATION DYNAMICS

The evolutionary population dynamics technique is based on self-organized criticality theory (SOC) [44], [45]. SOC theory indicates that the local changes in the population may influence the entire population without the intervention of any external [46]. The EPD aims to improve the quality of the solutions by removing the worst solutions at the current iteration of the solution and replacing the worst solution by generating new individuals around the best solution. EPD has been successfully incorporated in popular MH algorithms, such as extremal optimization (EO) [47] and evolutionary programming with self-organizing criticality (EPSCO) [48].

In [49] and [50], the EPD technique is applied to a single population. In this paper, we proposed a modified EPD.

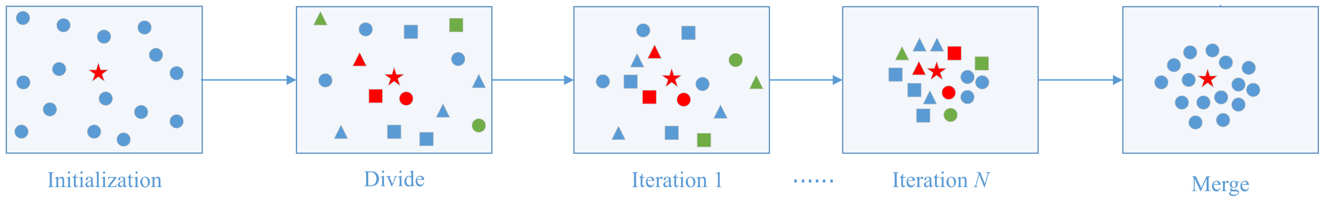


FIGURE 1. **The overall steps of EPD.**

We separate the population into three subpopulations, and each subpopulation evolves separately. In this way, we can improve the diversity of the population and improve the quality of the solution. After each iteration, the EPD technique updates the worst individual preventing interference from the worst individual on the present subpopulation. The primary process of EPD is shown in Figure 1.

IV. THE PROPOSED ALGORITHM

In this section, the structure of the proposed algorithm is explained. Algorithm 1 shows the main steps of the proposed algorithm. The flowchart of DECEHGS is presented in Figure 2.

In order to avoid the phenomenon of decreasing diversity when the population iterates to a specific region, the DE algorithm is adopted to maintain the diversity of the population. Moreover, the worst solution to the population may influence the algorithm's performance and prevent it from converging on the global optimum. Therefore, we integrate the updated solutions with the concept of CLS and EPD to avoid premature convergence and stagnation in the local optimum. The EPD technique replaces the worst agents by generating new agents in the neighborhood of the better ones. The CLS strategy creates a new agent randomly and chaotically in the neighborhood of the old agent.

1) INITIAL STAGE

The objective of this stage aims to produce the initial population X using the following equation.

$$X_i = lb_i + (ub_i - lb_i) \cdot rand \tag{16}$$

where lb_i and ub_i are lower and upper bounds of the search space, and i represents the current individual.

2) UPDATING POPULATION USING DE

The current solution X_i will be updated either using HGS or DE according to the fixed parameter L. For example, if a random number $r_1 < L$ (where $r_1 \in [0, 1]$), then the DE algorithm is applied to update the current solution X_i by Eq. (13). Whereas, if the $r_1 \geq L$ then the X_i will be calculated by Eq. (2). The mathematical formulation of this stage is shown as follows:

$$X_i = \begin{cases} X_i + F \cdot (X_{r_3} - X_{r_4}), & r_1 < L \\ W_1 \cdot X_b + R \cdot W_2 \cdot |X_b - X_i|, & r_1 > L,\ r_2 > E \\ W_1 \cdot X_b - R \cdot W_2 \cdot |X_b - X_i|, & r_1 > L,\ r_2 < E \end{cases} \tag{17}$$

where r_1 and $r_2 \in [0, 1]$, X_{r3} and X_{r4} are randomly selected from the main population, F is the scaling factory, W_1 and W_2 represent the hunger weight, R is a ranging controller, X_b is the best individual in the whole population and E is a variation control for all positions.

3) UPDATING POPULATION USING CLS

CLS strategy can avoid falling into the local optimum and improve the quality of the solution owing to its randomness and ergodicity. The equation of CLS is defined as:

$$X_i = (1 - L) \cdot X_i + L \cdot \beta_i \tag{18}$$

β_i is calculated by the following equation:

$$\beta_i = \mu\beta_i(1-\beta_i),\ \ \beta_i \in (0, 1),\ \beta_i \neq 0.25, 0.5 \textit{ and } 0.75 \tag{19}$$

When $\mu = 4$, the logistic function comes into a thorough chaotic state.

L is the contraction factor, which is determined as follows [51]:

$$L = 1 - |(t - 1)/t|^m \tag{20}$$

where t represents the current iteration and m controls the shrinking speed, the higher m value, the slower the shrinking rate.

4) UPDATING POPULATION USING EPD

The swarm of animals is divided into three subpopulations randomly to equip the HGS algorithm with the EPD technique. The worst animals are eliminated and reinitialized based on the good animals of each subpopulation. The formula for EPD is defined as follows:

$$X_i^c = rand \times (ub_i - lb_i) + lb, \quad i = 1, 2, \ldots, N_i \tag{21}$$

where ub_i and lb_i represent the boundaries of the search space of the best solution X_b, *rand* is a random number in the range of [0, 1], c indicates the current subpopulation, N_i represents the number of the worst agents, and its formula is defined as follows:

$$\begin{aligned} N_i &= round(N \times (rand \times (c_1 - c_2))), \\ c_1 &= 0.1, c_2 = 0.9 \end{aligned} \tag{22}$$

where *rand* is a random number in the interval [0, 1], N is the swarm size of the subpopulation, and *round()* is a function used to convert natural numbers to integers.

5) TERINATION CRITERION

The steps of the 2), 3), and 4) stages repeat until the stopping condition is satisfied. The number of fitness evaluations (*FES*) is used as a termination condition in this paper. When the DECEHGS reaches the stop condition, the best solution X_b is returned.

6) COMPUTATIONAL COMPLEXITY OF DECEHGS

The population size and subpopulation size are set to be N and SN, respectively. The dimension of the problem is set to be D, the maximum number of fitness evaluations is set to be T, and the swarm size of the worst individuals is set to be N_w. The complexity of DECEHGS depends on the complexity of HGS, DE, EPD, and CLS. So, the complexity of DECEHGS is defined as:

$$O(DEceHGS) = O(HGS) + O(DE) + O(EPD) + O(CLS) \tag{23}$$

where

$$\begin{aligned} O(HGS) &= O(N \times (1 + T \times SN \times (2 + \log SN + 2 \times D))) \\ O(DE) &= O(T \times SN \times D) \\ O(EPD) &= O(T \times N_w) \\ O(CLS) &= O(T \times D) \\ &So, \\ O(DEceHGS) &= O(T \times (N \times SN \times (\log SN + 3D + 3) + N_w) \\ &\quad + N \log SN + N \times D) \end{aligned} \tag{24}$$

V. EXPERIMENTAL STUDIES

In this section, comprehensive experiments were performed to validate the effectiveness of the proposed DECEHGS. All experiments were carried out using python 3.6 under Windows 10 with Intel (R) Core (TM) i5-1135G7 @ 2.40GHz and 16.0GB of RAM.

A. THE EFFECTIVENESS OF DE, CLS, AND EPD

The DECEHGS algorithm consists of three improved methods: the operators of DE, CLS strategy, and EPD technique. This section mainly aims to verify the effectiveness of these methods. We compared DECEHGS and its variants on the 30-dimensional CEC2014 test suites. The variants of which are DECHGS, DEEHGS, EHGS, CHGS, CEHGS, and DEHGS. The classic HGS algorithm uses DE and CLS strategies, but without the EPD technique, it is named DECHGS.

Algorithm 1 The Proposed DECEHGS Algorithm

Inputs: The population size N and maximum number of fitness evaluations *MAX_FES*
Outputs: The best individual and its fitness value
Initialize the population $X_i(i = 1, 2, \ldots, N)$
While (stopping condition is not met) do
  ## Divide
  Divide the main population X into three subpopulations
  ## Each subpopulation evolves
  For each subpopulation
    Calculate the fitness of each individual $F(i)$ and find the best individual X_b
    Update the parameters of *BF*, *WF*, *and SHungry*
    Use Eq. (9) to calculate the parameter of *Hungry*
    Use Eq. (7) and Eq. (8) to calculate the parameters of W_1 and W_2
    For each individual
      Use Eq. (3) and Eq. (5) to calculate the parameters of E and R
      Use Eq. (2) or Eq. (14) to update the current positions by parameter E
    End For
    Apply the EPD technique and CLS strategy by Eq. (18) and Eq. (21)
  End For
  ## Merge
  Three subpopulations are merged into one main population
End while
Return the best individual and its fitness value

TABLE 1. Parameter settings of involved algorithms.

Algorithm	Population size	Parameter settings
OBWOA	$N = 30$	$a_1 = [2,0]; a_2 = [-2,1]; b = 1$
TVBSSA	$N = 30$	$c_1 \in [0,1]; c_2 \in [0,1]$
ISCA	$N = 30$	$\alpha = 2$, *elite numbers* $= 2$
LSHcEpS	$N = 30$	$ps = 0.5; pc = 0.4; H = 5;$ $freq = 0.5; \mu F = 0.5; \mu CR = 0.5$
ILSHADE	$N = 30$	$p = 0.11; \mu F = 0.5; \mu CR = 0.9;$ $H = 6$
JSO	$N = 30$	$p_{max} = 0.25; p_{min} = p_{max}/2; H=5;$ $\mu F = 0.3; \mu CR = 0.9$
CoDE	$N = 30$	$\mu F \in [0.4, 0.9]; \mu CR \in [0.1, 0.9]$
BBPSO	$N = 30$	/
MGFPA	$N = 30$	$\gamma = 0.01; p = 0.8$
HIWOA	$N = 30$	$feedbackMax = 10; b = 1$ $p = 0.5; a \in [2, 0]; w \in [0.5, 1]$
mGWO	$N = 30$	$\mu CR = 0.5; a \in [2, 0]; k \in [1, 0]$
ESSA	$N = 30$	$r_1 \in [0, 50]; r_2 \in [0, 1]$
PPSO	$N = 30$	$c_1 = 2; c_2 = 2; wMax = 0.9;$ $wMin = 0.2; \theta \in [0, 2\pi]$
HGS	$N=30$	$LH = 10000; L = 0.08$

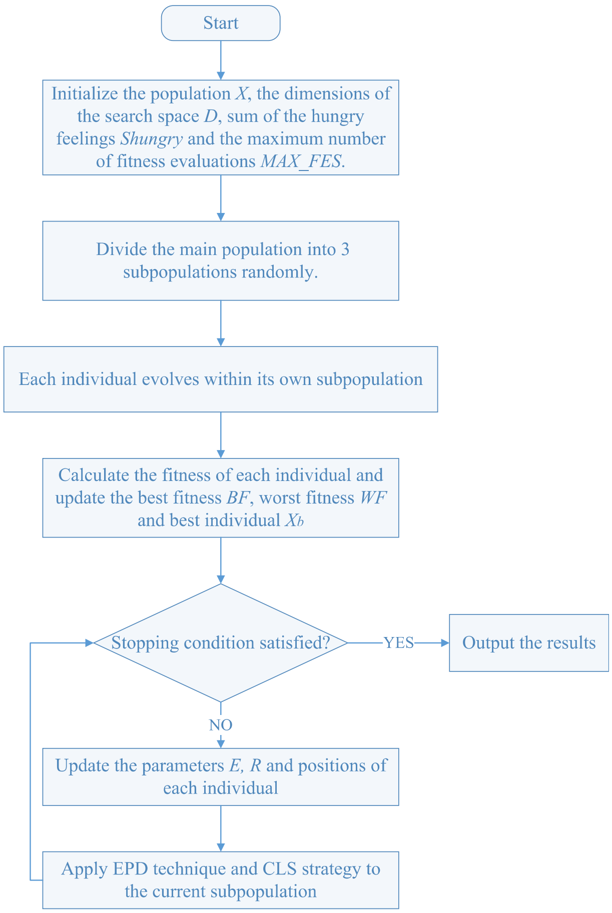


FIGURE 2. The main flowchart of the proposed DECEHGS algorithm.

The CEHGS algorithm uses only the CLS strategy and EPD technique. EHGS, CHGS, and DEHGS use only three improvement methods: EPD, CLS, and DE algorithm.

For the fairness of tests, all the involved algorithms were tested under the same conditions. The population size N and parameter settings are shown in Table 1. The maximum number of fitness evaluation *MAX_FES* was set to 1E5. It is noteworthy that each algorithm was executed according to the average results over 30 runs to reduce stochastic error in this paper.

The numerical results of these algorithms in terms of the average value (AVG), standard deviation (STD) of the function error rates, and rank of the algorithms (RANK) were obtained to assess the potentials of related techniques, and the best results of each task are marked in **boldface**.

It can be seen from Table 2 that among these seven HGS variants and the original HGS algorithm, DECEHGS ranks 1. The results show that DECEHGS has reached the global optimum value on F26, F29, and F30, but HGS has not achieved the global optimum value. CLS strategy can avoid local optimum and speed up the algorithm's convergence, and the EPD technique can enhance the performance of this algorithm. CEHGS successfully combines the advantages of these strategies and can reach the global optimum value of complex functions. On F1 to F9, DECEHGS, DECHGS, and DEEHGS are better than other variants on the test functions. This is because the DE algorithm is added, which can improve the diversity of the population allow the algorithms to find the global optimum. In summary, each component can enhance the performance of the original HGS, and our proposed algorithm can improve the performance of the original HGS much better by taking advantage of the three components (DE, CLS, and EPD).

B. COMPARISON WITH ADVANCED ALGORITHMS ON IEEE CEC2014 TEST SUITES

To further study the effectiveness of the proposed DECEHGS, we compared DECEHGS with improved L-SHADE algorithm (ILSHADE) [52], ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood (LSHADE_cnEpSi) (named as LSHcEpS) [53], DE with single objective real-parameter (JSO) [54], composite differential evolution (CoDE) [55], Bare bones particle swarm optimization (BBPSO) [56], improved sine cosine algorithm (ISCA) [57], opposition-based whale optimization algorithm (OBWOA) [58] and time-varying hierarchical salp swarm algorithm (TVBSSA) [59].

The experiment was conducted under the same conditions as the previous experiment to make the comparison results more impartial. The parameter settings for the involved algorithms are shown in Table 1. The maximum number of fitness evaluation *MAX_FES* was set to 1E5. Table 3 shows the average value, standard deviation, and rank of DECEHGS and other well-known algorithms on 30-dimensional CEC2014 test suites. As shown in Table 3, the DECEHGS algorithm has achieved minimum optimization on 15 test functions which shows great competitiveness over these advanced algorithms on CEC2014 test suites. The overall rank of DECEHGS is first among these combinations of DE, CLS, and EPD. EPD eliminated the worst solution and reinitialized the solution around the current best solution, DE algorithm increases the diversity of the population, and the CLS strategy can accelerate the algorithm's convergence rate.

The convergence curves of the DECEHGS and other advanced algorithms are shown in Figure 3. It can be observed that the DECEHGS converges faster than other advanced algorithms in most cases. On F1, F3, F4, F7, F11, and F13, the convergence rate of DECEHGS is fast, DECEHGS has not reached the global optimum, but the solution accuracy is very high. Through the convergence graphs of F14, F16, F19, F23, and F27, the convergence rate of DECEHGS is not the fastest, but the final optimization results of DECEHGS are much smaller than other competitors. On F24 to F26 and F28 to F30, DECEHGS has the quickest convergence rate and can find the global optimum, and other advanced algorithms have fallen into local optimum.

C. VALIDATION ON IEEE CEC2017 TEST SUITES WITH DIFFERENT DIMENSIONS

In this part, to further verify the scalability of the proposed algorithm, DECEHGS was tested with different dimensional

TABLE 2. Comparison results of DECEHGS, DECHGS, DEEHGS, EHGS, CHGS, CEHGS, DEHGS, and HGS algorithms on CEC2014 test suites.

Function		DECEHGS	DECHGS	DEEHGS	EHGS	CHGS	CEHGS	DEHGS	HGS
	AVG	**3.15E+05**	1.78E+06	1.08E+06	6.40E+06	4.07E+06	3.51E+06	3.23E+06	5.25E+06
F1	STD	**5.28E+05**	1.04E+06	6.92E+05	4.46E+06	2.40E+06	1.88E+06	2.36E+06	3.00E+06
	RANK	**1**	3	2	8	6	5	4	7
	AVG	9.56E+02	1.33E+04	**6.19E+02**	7.85E+02	1.35E+04	6.83E+02	1.42E+04	1.40E+04
F2	STD	2.06E+03	1.16E+04	7.56E+02	1.60E+03	1.04E+04	**7.23E+02**	1.31E+04	1.24E+04
	RANK	4	5	**1**	3	6	2	8	7
	AVG	1.12E+03	4.40E+03	**9.08E+02**	3.46E+03	6.49E+03	4.06E+03	5.49E+03	7.29E+03
F3	STD	**6.56E+02**	2.35E+03	6.73E+02	2.45E+03	3.17E+03	3.64E+03	2.54E+03	4.68E+03
	RANK	2	5	**1**	3	7	4	6	8
	AVG	4.85E+02	4.98E+02	4.96E+02	4.86E+02	4.97E+02	**4.84E+02**	5.00E+02	5.11E+02
F4	STD	**2.48E+01**	3.51E+01	3.42E+01	3.69E+01	3.33E+01	3.14E+01	3.27E+01	3.98E+01
	RANK	2	6	4	3	5	**1**	7	8
	AVG	5.20E+02	**5.20E+02**	5.20E+02	5.20E+02	5.20E+02	5.20E+02	5.20E+02	5.20E+02
F5	STD	1.69E-01	**1.01E-01**	1.87E-01	1.31E-01	1.36E-01	1.52E-01	1.12E-01	1.14E-01
	RANK	5	**1**	8	7	3	6	4	2
	AVG	**6.25E+02**	6.27E+02	6.30E+02	6.28E+02	6.29E+02	6.25E+02	6.29E+02	6.30E+02
F6	STD	4.40E+00	3.71E+00	4.25E+00	3.94E+00	3.78E+00	3.54E+00	**3.21E+00**	3.57E+00
	RANK	**1**	3	7	4	6	2	5	8
	AVG	**7.00E+02**	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.01E+02
F7	STD	**4.16E-03**	3.80E-02	2.26E-02	3.49E-02	2.98E-01	3.67E-02	2.49E-01	2.73E+00
	RANK	**1**	5	2	3	7	4	6	8
	AVG	9.29E+02	**8.94E+02**	9.19E+02	9.09E+02	9.06E+02	9.22E+02	8.97E+02	9.40E+02
F8	STD	2.50E+01	**1.90E+01**	3.52E+01	2.34E+01	1.89E+01	2.42E+01	2.08E+01	3.63E+01
	RANK	7	**1**	5	4	3	6	2	8
	AVG	1.07E+03	1.08E+03	1.08E+03	**1.07E+03**	1.08E+03	1.07E+03	1.08E+03	1.08E+03
F9	STD	2.17E+01	2.39E+01	2.22E+01	2.57E+01	2.24E+01	**1.75E+01**	2.31E+01	2.63E+01
	RANK	3	7	6	**1**	8	2	5	4
	AVG	3.83E+03	3.30E+03	4.40E+03	4.18E+03	**3.19E+03**	3.51E+03	3.46E+03	4.26E+03
F10	STD	6.83E+02	6.79E+02	8.20E+02	7.52E+02	5.90E+02	7.56E+02	**5.80E+02**	7.05E+02
	RANK	5	2	8	6	**1**	4	3	7
	AVG	5.37E+03	**5.04E+03**	5.61E+03	5.30E+03	5.36E+03	5.15E+03	5.27E+03	5.33E+03
F11	STD	9.73E+02	5.88E+02	5.98E+02	5.86E+02	5.93E+02	7.55E+02	**5.60E+02**	7.57E+02
	RANK	7	**1**	8	4	6	2	3	5
	AVG	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03
F12	STD	3.66E-01	3.07E-01	4.86E-01	3.08E-01	3.45E-01	**2.23E-01**	2.62E-01	3.18E-01
	RANK	5	4	8	6	7	**1**	2	3
	AVG	**1.30E+03**	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03
F13	STD	**9.87E-02**	1.07E-01	1.14E-01	1.20E-01	1.13E-01	1.08E-01	1.56E-01	1.46E-01
	RANK	**1**	6	4	3	5	2	8	7
	AVG	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	**1.40E+03**	1.40E+03	1.40E+03
F14	STD	1.74E-01	1.58E-01	2.05E-01	1.84E-01	2.08E-01	**4.58E-02**	2.69E-01	2.59E-01
	RANK	2	4	6	5	3	**1**	8	7
	AVG	1.52E+03	1.54E+03	1.52E+03	**1.52E+03**	1.54E+03	1.52E+03	1.53E+03	1.55E+03
F15	STD	1.09E+01	1.28E+01	9.11E+00	**5.78E+00**	1.42E+01	6.09E+00	1.41E+01	3.98E+01
	RANK	4	6	2	**1**	7	3	5	8
	AVG	1.61E+03	**1.54E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F16	STD	6.89E-01	1.28E+01	4.97E-01	4.16E-01	3.80E-01	5.04E-01	4.26E-01	**3.30E-01**
	RANK	2	**1**	4	5	6	3	7	8
	AVG	**1.71E+05**	2.37E+05	4.15E+05	7.84E+05	4.34E+05	5.14E+05	3.76E+05	8.19E+05
F17	STD	**8.80E+04**	1.19E+05	2.58E+05	7.62E+05	2.30E+05	3.44E+05	2.32E+05	4.53E+05
	RANK	**1**	2	4	7	5	6	3	8
	AVG	4.09E+03	6.36E+03	6.40E+03	7.70E+03	4.19E+03	**3.45E+03**	9.08E+03	7.89E+03
F18	STD	3.18E+03	6.74E+03	5.30E+03	5.53E+03	2.40E+03	**2.09E+03**	7.58E+03	6.59E+03
	RANK	2	4	5	6	3	**1**	8	7
	AVG	**1.91E+03**	1.92E+03	1.92E+03	1.93E+03	1.94E+03	1.91E+03	1.92E+03	1.92E+03
F19	STD	**1.01E+01**	1.76E+01	1.58E+01	3.05E+01	3.72E+01	1.03E+01	2.47E+01	1.81E+01
	RANK	**1**	4	3	7	8	2	6	5
	AVG	9.47E+03	1.12E+04	**7.85E+03**	1.07E+04	1.26E+04	1.02E+04	9.82E+03	2.14E+04
F20	STD	6.32E+03	4.64E+03	**3.52E+03**	4.65E+03	4.92E+03	7.17E+03	4.81E+03	1.07E+04
	RANK	2	6	**1**	5	7	4	3	8
	AVG	**7.56E+04**	2.40E+05	1.27E+05	2.78E+05	2.43E+05	2.20E+05	1.86E+05	4.01E+05
F21	STD	**5.77E+04**	2.32E+05	8.60E+04	2.44E+05	1.96E+05	2.22E+05	1.89E+05	3.54E+05
	RANK	**1**	5	2	7	6	4	3	8
	AVG	2.76E+03	2.83E+03	2.79E+03	**2.68E+03**	2.87E+03	2.68E+03	2.84E+03	2.91E+03
F22	STD	1.71E+02	2.60E+02	2.30E+02	1.66E+02	2.11E+02	**1.31E+02**	1.82E+02	2.37E+02
	RANK	3	5	4	**1**	7	2	6	8
	AVG	**2.50E+03**	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03
F23	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RANK	**1**	1	1	1	1	1	1	1

TABLE 2. ***(Continued.)*** **Comparison results of DECEHGS, DECHGS, DEEHGS, EHGS, CHGS, CEHGS, DEHGS, and HGS algorithms on CEC2014 test suites.**

F24	AVG	**2.60E+03**	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03
	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RANK	**1**	1	1	1	1	1	1	1
F25	AVG	**2.70E+03**	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03
	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RANK	**1**	1	1	1	1	1	1	1
F26	AVG	**2.70E+03**	2.73E+03	2.70E+03	2.71E+03	2.71E+03	2.71E+03	2.72E+03	2.73E+03
	STD	4.69E+01	4.79E+01	1.79E+01	2.48E+01	2.97E+01	**2.48E+01**	4.20E+01	4.56E+01
	RANK	**1**	8	2	3	5	4	6	7
F27	AVG	**2.90E+03**	2.90E+03	2.90E+03	2.90E+03	2.90E+03	2.90E+03	2.90E+03	2.90E+03
	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RANK	**1**	1	1	1	1	1	1	1
F28	AVG	**3.00E+03**	3.00E+03	3.00E+03	3.00E+03	3.00E+03	3.00E+03	3.00E+03	3.00E+03
	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RANK	**1**	1	1	1	1	1	1	1
F29	AVG	**3.10E+03**	3.10E+03	3.66E+03	4.08E+03	3.10E+03	3.10E+03	4.38E+03	5.73E+05
	STD	**0.00E+00**	0.00E+00	1.27E+03	1.33E+03	0.00E+00	0.00E+00	1.06E+03	2.13E+06
	RANK	**1**	1	5	6	1	1	7	8
F30	AVG	**3.20E+03**	3.20E+03	6.56E+03	1.19E+04	3.20E+03	3.20E+03	6.45E+03	1.05E+04
	STD	**0.00E+00**	0.00E+00	6.13E+03	1.01E+04	0.00E+00	0.00E+00	2.95E+03	6.58E+03
	RANK	**1**	1	6	8	1	1	5	7
Sum of ranks		**70**	101	113	121	134	78	135	176
Average rank		**2.33**	3.37	3.77	4.03	4.47	2.60	4.50	5.87
Overall rank		1	3	4	5	6	2	7	8

problems (i.e., $Dim = 10$, 30, 50, and 100). The experimental results obtained from CEC2017 test suites are presented in Table 4. Some state-of-the-art algorithms are employed to test the performance of DECEHGS, such as enhanced salp swarm algorithm (ESSA) [60], modified global flower pollination algorithm (MGFPA) [61], memory-based grey wolf optimizer (mGWO) [3], hybrid improved whale optimization algorithm (HIWOA) [62] and phasor particle swarm optimization (PPSO) [63]. In this experiment, the population size N and parameter settings are shown in Table 1, the maximum number of fitness evaluations was set to 1E5. Each algorithm was executed 30 times randomly.

Table 4 shows the comparison between DECEHGS and the advanced algorithms with different dimensions. As shown in Table 4, DECEHGS ranked first across different dimensions. For unimodal functions (F1 to F3), DECEHGS converges very quickly, and the quality of its solution is very high. For multi-modal functions, DECEHGS performs better at higher dimensions. It is worth noting that for composition functions (F23 to F30), the performance of DECEHGS did not deteriorate seriously as the dimension increased. These results indicate that the searchability of DECEHGS is effective. Moreover, DECEHGS can avoid falling into local optimum, and the optimization performance of solving high dimensional functions is strong.

The convergence curves of DECEHGS and other algorithms on CEC2017 test suites are shown in Figure 4. In Figure 4, F1 and F3 are unimodal functions, F4, F5, F8 to F12, and F16 are multi-modal functions, F17 is a hybrid function, and F27 to F30 are composition functions. The first row of the figure shows the results of the involved algorithms on a 10-dimensional test set. The second row displays the results of the 30-dimensional test set. The third row represents the results of 50-dimensional benchmark functions. The graph in the fourth row shows the results of the 100-dimensional test set.

It is not difficult to see from Figure 4 that of these advanced algorithms, DECEHGS has the best optimization effect. As shown in Figure 4, DECEHGS converges very fast and outperforms most of the other algorithms. In dealing with F17 and F29, there is close competition among all algorithms, but DECEHGS can find a better solution during the overall steps. Although PPSO's convergence rate is faster than DECEHGS on F29, PPSO's optimization effect is worse than DECEHGS. It is because the EPD technique in DECEHGS helps the population deviate from the local optimum. The DE algorithm can increase the diversity of the population, and the CLS strategy can accelerate the algorithm's convergence rate. All in all, the experimental results of DECEHGS are superior to other advanced algorithms.

D. ENGINEERING PROBLEMS

The proposed algorithm was validated using four engineering optimization problems: the welded beam design problem [64], tension/compression spring design problem [65], pressure vessel design problem, and three bar truss design problem. The proposed methods were implemented for 30 independent runs. The population size N and parameter settings were set the same as the original paper, and the maximum number of fitness evaluation MAX_FES is set to 1E4. The results are shown in Table 5, Table 6, Table 7, and Table 8.

The penalty function approach was utilized to handle optimization constraints [66], [67]. The formula is stated as:

$$\min \phi(x) = f(x) + \lambda \sum_{c \in N_e} (\max(0, g_e(x)))^2 \tag{25}$$

TABLE 3. Comparison results of DECEHGS and other advanced algorithms on CEC2014 test suites.

Function		DECEHGS	ILSHADE	JSO	LSHcEpS	CoDE	BBPSO	ISCA	OBWOA	TVBSSA
	AVG	**3.15E+05**	4.81E+06	5.86E+06	1.90E+07	1.97E+07	3.30E+06	1.27E+08	2.06E+08	7.41E+07
F1	STD	**5.28E+05**	2.84E+06	3.93E+06	1.15E+07	1.43E+07	2.79E+06	6.50E+07	1.22E+08	2.35E+07
	RANK	**1**	3	4	5	6	2	8	9	7
	AVG	9.56E+02	9.11E+08	2.23E+08	7.21E+08	**2.00E+02**	3.03E+02	9.10E+09	4.20E+09	4.84E+08
F2	STD	**2.06E+03**	1.27E+09	5.65E+08	1.55E+09	1.97E+06	2.31E+02	4.83E+09	3.04E+09	1.19E+08
	RANK	3	7	4	6	**1**	2	9	8	5
	AVG	**1.12E+03**	4.05E+03	3.74E+03	3.73E+03	1.80E+03	2.16E+03	4.32E+04	7.55E+04	6.31E+04
F3	STD	**6.56E+02**	2.16E+03	2.36E+03	2.99E+03	1.17E+03	1.98E+03	9.75E+03	3.72E+04	1.35E+04
	RANK	**1**	6	5	4	2	3	7	9	8
	AVG	**4.85E+02**	5.35E+02	5.39E+02	6.32E+02	5.10E+02	4.96E+02	1.00E+03	1.28E+03	5.63E+02
F4	STD	**2.48E+01**	4.06E+01	3.88E+01	5.04E+01	3.78E+01	3.41E+01	3.73E+02	3.19E+02	4.97E+01
	RANK	**1**	4	5	7	3	2	8	9	6
	AVG	5.20E+02	5.20E+02	5.20E+02	**5.20E+02**	5.20E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F5	STD	1.69E-01	7.36E-02	1.08E-01	**1.31E-02**	1.43E-01	5.69E-02	5.15E-02	9.41E-02	5.39E-02
	RANK	4	2	3	**1**	5	8	7	6	9
	AVG	6.25E+02	6.40E+02	6.39E+02	6.41E+02	6.23E+02	**6.20E+02**	6.24E+02	6.38E+02	6.28E+02
F6	STD	4.40E+00	3.78E+00	**3.25E+00**	2.33E+00	6.92E+00	5.47E+00	2.58E+00	4.03E+00	4.03E+00
	RANK	4	8	7	9	2	**1**	3	6	5
	AVG	**7.00E+02**	7.06E+02	7.02E+02	7.08E+02	7.00E+02	7.00E+02	7.79E+02	7.11E+02	7.05E+02
F7	STD	**4.16E-03**	6.87E+00	3.21E+00	1.46E+01	5.38E-02	1.12E-02	3.29E+01	3.25E+00	1.03E+00
	RANK	**1**	6	4	7	3	2	9	8	5
	AVG	9.29E+02	9.66E+02	9.61E+02	9.67E+02	**8.01E+02**	8.70E+02	9.38E+02	1.03E+03	1.03E+03
F8	STD	2.50E+01	7.72E+00	7.38E+00	6.21E+00	**3.67E+00**	1.55E+01	2.40E+01	3.34E+01	3.45E+01
	RANK	3	6	5	7	**1**	2	4	8	9
	AVG	1.07E+03	1.09E+03	1.09E+03	1.09E+03	1.05E+03	**1.03E+03**	1.07E+03	1.19E+03	1.15E+03
F9	STD	2.17E+01	7.03E+00	6.21E+00	4.69E+00	**9.89E+00**	3.73E+01	2.14E+01	5.31E+01	4.13E+01
	RANK	3	6	7	5	2	**1**	4	9	8
	AVG	3.83E+03	5.16E+03	4.98E+03	5.20E+03	**1.10E+03**	2.29E+03	4.85E+03	5.66E+03	6.41E+03
F10	STD	6.83E+02	3.18E+02	3.43E+02	2.77E+02	**1.38E+02**	3.39E+02	5.79E+02	7.49E+02	7.59E+02
	RANK	3	6	5	7	**1**	2	4	8	9
	AVG	**5.37E+03**	5.38E+03	5.48E+03	5.44E+03	5.88E+03	5.73E+03	6.47E+03	7.59E+03	7.01E+03
F11	STD	**9.73E+02**	2.85E+02	2.61E+02	2.48E+02	3.18E+02	2.08E+03	6.86E+02	6.46E+02	6.39E+02
	RANK	**1**	2	4	3	6	5	7	9	8
	AVG	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03
F12	STD	3.66E-01	9.76E-02	**4.49E-02**	7.79E-02	1.63E-01	6.85E-01	4.03E-01	4.40E-01	4.35E-01
	RANK	4	3	**1**	2	5	7	8	6	9
	AVG	**1.30E+03**	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03
F13	STD	**9.87E-02**	4.83E-01	1.28E-01	1.49E-01	6.70E-02	1.23E-01	9.16E-01	4.53E-01	1.03E-01
	RANK	**1**	6	4	5	3	2	9	8	7
	AVG	**1.40E+03**	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.42E+03	1.41E+03	1.40E+03
F14	STD	**1.74E-01**	1.30E+00	1.30E-01	5.93E+00	7.45E-02	2.11E-01	9.24E+00	5.46E+00	2.20E-01
	RANK	**1**	6	3	7	2	4	9	8	5
	AVG	1.52E+03	1.59E+03	1.53E+03	1.90E+03	1.52E+03	**1.51E+03**	3.32E+03	3.02E+03	1.53E+03
F15	STD	1.09E+01	7.16E+01	1.06E+01	4.60E+02	**3.16E+00**	3.75E+00	2.74E+03	1.38E+03	6.21E+00
	RANK	3	6	4	7	2	**1**	9	8	5
	AVG	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F16	STD	6.89E-01	6.41E-02	9.25E-02	2.63E-01	**1.86E-01**	6.74E-01	3.49E-01	4.19E-01	4.12E-01
	RANK	**1**	9	8	7	2	3	6	4	5
	AVG	1.71E+05	1.16E+05	7.92E+04	7.28E+05	**1.13E+04**	7.57E+05	2.68E+06	1.90E+07	2.87E+06
F17	STD	8.80E+04	8.99E+04	**7.48E+04**	7.11E+05	8.19E+05	5.07E+05	1.44E+06	1.57E+07	1.98E+06
	RANK	4	3	2	5	**1**	6	7	9	8
	AVG	4.09E+03	5.91E+06	3.17E+03	**2.79E+03**	4.12E+04	9.52E+03	3.64E+07	2.86E+06	1.76E+06
F18	STD	3.18E+03	3.18E+07	2.67E+03	**9.30E+02**	2.06E+05	1.57E+04	5.17E+07	8.23E+06	1.17E+06
	RANK	3	8	2	**1**	5	4	9	7	6
	AVG	**1.91E+03**	1.93E+03	1.92E+03	1.95E+03	1.92E+03	1.92E+03	1.97E+03	2.01E+03	1.92E+03
F19	STD	1.01E+01	2.51E+01	3.36E+01	3.87E+01	1.13E+00	1.20E+01	2.65E+01	5.10E+01	**2.48E+00**
	RANK	**1**	6	5	7	2	3	8	9	4
	AVG	9.47E+03	6.21E+03	6.23E+03	7.66E+03	**2.08E+03**	1.71E+04	2.52E+04	8.18E+04	5.74E+04
F20	STD	6.32E+03	2.64E+03	3.76E+03	3.60E+03	**2.45E+02**	1.39E+04	1.25E+04	8.27E+04	3.84E+04
	RANK	5	2	3	4	**1**	6	7	9	8
	AVG	7.56E+04	7.36E+04	2.22E+04	1.31E+05	**3.62E+03**	2.87E+05	7.90E+05	4.77E+06	1.31E+06
F21	STD	5.77E+04	9.01E+04	2.33E+04	1.42E+05	**1.89E+04**	2.34E+05	1.00E+06	3.87E+06	8.23E+05
	RANK	4	3	2	5	**1**	6	7	9	8
	AVG	2.76E+03	2.82E+03	2.60E+03	3.16E+03	**2.41E+03**	2.69E+03	2.62E+03	3.18E+03	2.96E+03
F22	STD	**1.71E+02**	3.02E+02	1.78E+02	2.81E+02	1.79E+02	2.23E+02	1.64E+02	2.39E+02	2.28E+02
	RANK	5	6	2	8	**1**	4	3	9	7
	AVG	**2.50E+03**	2.50E+03	2.50E+03	2.50E+03	2.62E+03	2.61E+03	2.65E+03	2.68E+03	2.65E+03
F23	STD	**0.00E+00**	2.39E+00	4.30E-01	0.00E+00	9.70E-02	1.82E-12	1.63E+01	1.73E+01	1.44E+01
	RANK	**1**	4	3	1	6	5	7	9	8

TABLE 3. *(Continued.)* **Comparison results of DECEHGS and other advanced algorithms on CEC2014 test suites.**

F24	AVG	**2.60E+03**	2.60E+03	2.60E+03	2.60E+03	2.63E+03	2.64E+03	2.60E+03	2.60E+03	2.62E+03
	STD	**5.39E-07**	3.41E-01	2.35E-01	1.66E-02	3.31E+00	8.06E+00	1.46E-02	5.55E+00	1.87E+01
	RANK	**1**	5	4	3	8	9	2	6	7
F25	AVG	**2.70E+03**	2.70E+03	2.70E+03	2.70E+03	2.71E+03	2.70E+03	2.72E+03	2.71E+03	2.71E+03
	STD	**0.00E+00**	2.84E-02	1.30E-02	0.00E+00	1.15E-09	4.09E-01	6.73E+00	1.65E+01	8.96E+00
	RANK	**1**	4	3	1	7	5	9	6	8
F26	AVG	**2.70E+03**	2.80E+03	2.80E+03	2.80E+03	2.70E+03	2.70E+03	2.70E+03	2.71E+03	2.70E+03
	STD	**4.69E+01**	2.45E-04	2.02E-04	1.62E-04	7.74E-02	1.08E-01	7.74E-01	2.98E+01	1.44E-01
	RANK	**1**	9	8	7	2	3	5	6	4
F27	AVG	**2.90E+03**	2.90E+03	2.90E+03	2.90E+03	3.55E+03	3.56E+03	3.59E+03	3.71E+03	3.64E+03
	STD	**0.00E+00**	1.66E+00	2.41E-01	0.00E+00	1.77E+02	1.75E+02	1.30E+02	3.64E+02	2.23E+02
	RANK	**1**	4	3	1	5	6	7	9	8
F28	AVG	**3.00E+03**	3.00E+03	3.00E+03	3.00E+03	3.67E+03	3.22E+03	4.29E+03	5.04E+03	3.45E+03
	STD	**0.00E+00**	7.59E-01	2.45E-01	0.00E+00	8.16E-01	2.40E+01	2.83E+02	8.17E+02	1.42E+02
	RANK	**1**	4	3	1	7	5	8	9	6
F29	AVG	**3.10E+03**	2.10E+06	3.49E+05	3.10E+03	4.36E+05	3.12E+03	7.65E+06	1.16E+07	3.14E+03
	STD	**0.00E+00**	1.99E+06	7.76E+05	0.00E+00	2.51E+06	2.25E+01	7.39E+06	6.23E+06	3.78E+01
	RANK	**1**	7	5	1	6	3	8	9	4
F30	AVG	**3.20E+03**	1.38E+05	3.79E+04	3.20E+03	5.58E+03	3.72E+03	1.52E+05	2.91E+05	4.55E+03
	STD	**0.00E+00**	1.52E+05	7.57E+04	4.82E-05	1.29E+03	1.99E+02	6.90E+04	1.68E+05	3.60E+02
	RANK	**1**	7	6	2	5	3	8	9	4
Sum of ranks		**65**	158	124	136	103	115	206	238	200
Average rank		**2.17**	5.27	4.13	4.53	3.43	3.83	6.87	7.93	6.67
Overall rank		**1**	6	4	5	2	3	8	9	7

where λ and N_e are the penalty coefficients and the number of constraints, respectively. $f(x)$ and $g_e(x)$ denote the objective function and their constraints, respectively.

1) WELDED BEAM DESIGN (WBD) PROBLEM

The main objective of designing welded beams is to find the lowest consumption of welded beams under the four constraints of bucking load (P_c), shear stress (τ), bending stress in the beam (θ), and deflection rate (δ). The problem includes four variables: welding thickness (h), length of steel joint (l), the thickness of steel (b), and height of steel (t). The mathematical model is defined as follows:

$$
\begin{aligned}
&consider: \\
&\quad x = [x_1, x_2, x_3, x_4] = [h, l, t, b] \\
&\min f(x) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\
&subject to: \\
&\quad g_1(x) = \tau(x) - \tau_{\max} \leq 0 \\
&\quad g_2(x) = \sigma(x) - \sigma\max \leq 0 \\
&\quad g_3(x) = x_1 - x_4 \leq 0 \\
&\quad g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \\
&\quad g_5(x) = 0.125 - x_1 \leq 0 \\
&\quad g_6(x) = \delta(x) - \delta_{\max} \leq 0 \\
&\quad g_7(x) = P - P_c(x) \leq 0 \\
&variable range: \\
&\quad 0.1 \leq x_i \leq 2, i = 1, 4 \\
&\quad 0.1 \leq x_i \leq 10, i = 2, 3 \\
&\quad \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + \tau''} \\
&\quad \tau' = \frac{p}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J} \\
&\quad M = P(L + \frac{x_2}{2}) \\
&\quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
&\quad P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x^3}{2L}\sqrt{\frac{E}{4G}}\right) \\
&\quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\
&\quad \sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4} \\
&\quad P = 6000lb, L = 14in, e = 30\times10^6psi, G = 12\times10^6 \\
&\quad \tau_max = 13,600psi, \sigma_{\max} = 30,000psi, \delta_{\max} = 0.25in
\end{aligned}
$$

On this subject, DECEHGS was compared with self-adaptive differential evolution algorithm (SADE) [68], success-history-based parameter adaptation for differential evolution (SHADE) [69], LSHcEpS, adaptive differential evolution algorithm with novel mutation strategies (MPADE) [70], GWO, GA, MVO, HHO, BBO, HS, sine cosine algorithm (SCA) [71], HGS and davidon-fletcher-powell (DAVID) [72].

The comparison results from Table 5 show that DECEHGS is the best algorithm for the WBD problem. Four parameters are set to be 0.20521, 3.470466, 9.036321, and 0.21524, respectively, and the manufacturing cost of WBD can reach 1.724745.

2) TENSION/COMPRESSION SPRING DESIGN (TCSD) PROBLEM

The tension/compression spring design problem is described in Arora, for which the aim is to determine the optimal values

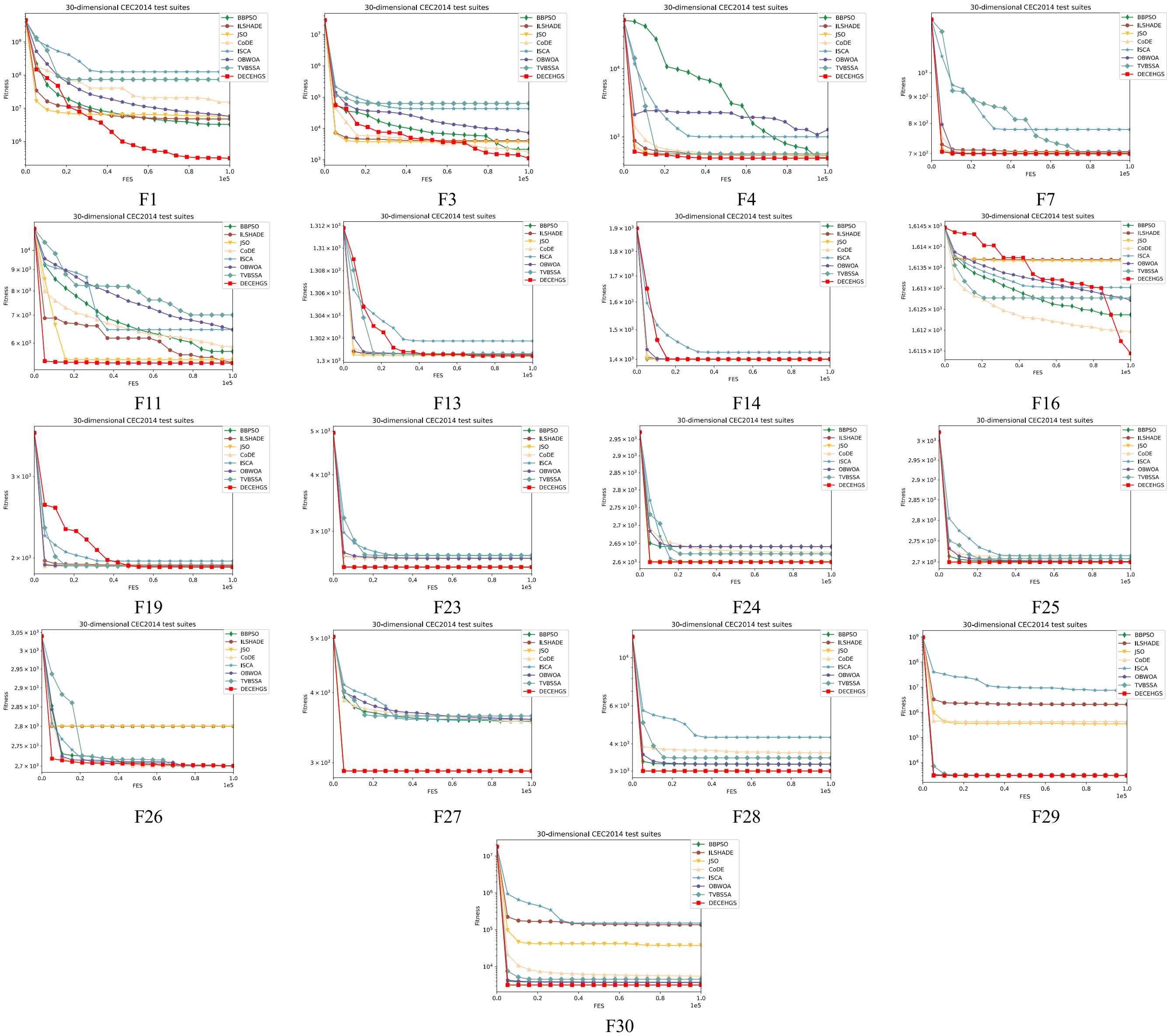


FIGURE 3. Convergence trends curves for DECEHGS versus other advanced algorithms on CEC2014 test suites.

of active coils (N), the wire diameter (d), and mean coil diameter (D). The mathematical model is as follows:

$$\begin{aligned} &\textit{consider}: \\ &\quad x = [x_1, x_2, x_3] = [d, D, N], \\ &\min f(x) = (x_3 + 2)x_2 x_1^2 \\ &\quad g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\ &\quad g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0 \\ &\quad g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\ &\quad g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{aligned}$$

$$\begin{aligned} &\textit{variables}\ \text{range}: \\ &\quad 0.05 \leq x_1 \leq 2 \\ &\quad 0.25 \leq x_2 \leq 1.30 \\ &\quad 2.00 \leq x_3 \leq 15 \end{aligned}$$

On this subject, the proposed algorithm is compared with other state-of-the-art techniques such as LSHcEpS, SHADE, a mathematical programming method by Belegundu and Arora [73] (named as BAA), BBO, moth flame optimization (MFO) [74], HGS, MVO, GA, GSA and MPADE methods.

It can be seen from Table 6 that DECEHGS and MPADE expose the best overall performance. The MFO, SHADE, HGS algorithms rank the 3nd, 4rd, and 5th positions, respectively.

TABLE 4. Comparison results of DECEHGS and other advanced algorithms on CEC2017 test suites with different dimensions.

Function		DECEHGS				MGFPA				ESSA			
Dim		*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100	*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100	*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100
	AVG	**2.30E+03**	**6.39E+03**	**3.97E+04**	**1.51E+06**	5.10E+10	4.85E+10	2.62E+11	1.30E+12	5.89E+08	4.46E+09	2.08E+10	1.61E+11
F1	STD	**1.94E+03**	**6.74E+03**	**3.38E+03**	**2.06E+06**	1.43E+10	2.24E+10	5.62E+10	1.06E+11	3.16E+09	2.74E+09	9.78E+09	2.08E+10
	RANK	**1**	**1**	**1**	**1**	5	5	5	5	4	4	4	4
	AVG	**3.00E+02**	2.71E+03	**3.71E+04**	2.46E+05	3.01E+02	**3.00E+02**	5.09E+04	**1.26E+05**	5.44E+04	5.52E+04	1.33E+05	3.10E+05
F3	STD	**5.98E-13**	1.52E+03	7.38E+03	2.40E+04	1.03E+00	**4.28E-01**	**1.78E+03**	2.23E+04	6.47E+03	6.56E+03	1.32E+04	1.14E+04
	RANK	**1**	2	**1**	2	2	**1**	2	**1**	5	5	5	3
	AVG	**4.01E+02**	**4.81E+02**	**5.25E+02**	**7.67E+02**	1.24E+03	1.33E+03	3.63E+03	1.74E+04	5.38E+02	6.19E+02	9.16E+02	2.28E+03
F4	STD	**1.98E-01**	**3.48E+01**	**4.56E+01**	**5.73E+01**	3.93E+02	4.01E+02	6.43E+02	4.24E+03	4.87E+01	6.81E+01	1.70E+02	4.44E+02
	RANK	**1**	**1**	**1**	**1**	6	5	5	5	4	4	4	4
	AVG	**5.24E+02**	**6.34E+02**	8.41E+02	1.34E+03	5.87E+02	6.68E+02	**7.30E+02**	**1.30E+03**	6.68E+02	6.65E+02	8.19E+02	1.37E+03
F5	STD	**1.08E+01**	3.96E+01	3.15E+01	3.18E+01	2.39E+01	2.54E+01	7.83E+01	1.89E+02	2.46E+01	3.46E+01	4.07E+01	5.25E+01
	RANK	**1**	**1**	5	2	3	5	**1**	**1**	5	4	2	3
	AVG	**6.09E+02**	6.60E+02	6.82E+02	6.77E+02	6.18E+02	6.20E+02	6.37E+02	6.65E+02	6.64E+02	6.66E+02	6.84E+02	6.84E+02
F6	STD	**8.52E+00**	1.11E+01	1.03E+01	3.21E+00	4.98E+00	6.02E+00	6.95E+00	5.63E+00	1.05E+01	7.85E+00	8.72E+00	6.59E+00
	RANK	**1**	4	4	3	3	2	2	2	5	5	5	5
	AVG	7.63E+02	1.06E+03	1.55E+03	2.96E+03	8.80E+02	**8.77E+02**	1.20E+03	2.54E+03	9.61E+02	9.58E+02	1.23E+03	**2.26E+03**
F7	STD	2.10E+01	7.09E+01	1.39E+02	1.97E+02	3.00E+01	**2.66E+01**	9.65E+01	2.42E+02	4.21E+01	4.74E+01	7.11E+01	**1.47E+02**
	RANK	2	5	5	5	3	**1**	2	3	5	4	3	**1**
	AVG	**8.26E+02**	**9.22E+02**	1.20E+03	1.85E+03	8.76E+02	9.61E+02	**1.02E+03**	**1.53E+03**	9.42E+02	9.38E+02	1.14E+03	1.68E+03
F8	STD	**1.28E+01**	3.32E+01	4.48E+01	9.29E+01	2.38E+01	2.17E+01	6.62E+01	**1.82E+02**	2.18E+01	2.58E+01	3.99E+01	8.58E+01
	RANK	**1**	**1**	4	4	3	4	**1**	**1**	4	3	2	3
	AVG	**9.27E+02**	4.56E+03	1.21E+04	**2.28E+04**	1.53E+03	1.59E+03	6.64E+03	3.63E+04	4.51E+03	4.44E+03	1.34E+04	3.54E+04
F9	STD	**4.00E+01**	6.60E+02	**1.22E+03**	**1.54E+03**	4.24E+02	3.80E+02	1.60E+03	5.26E+03	8.21E+02	9.83E+02	2.18E+03	3.33E+03
	RANK	**1**	5	3	**1**	4	2	2	5	5	4	5	4
	AVG	1.94E+03	5.16E+03	**6.18E+03**	**1.50E+04**	6.51E+03	6.81E+03	1.28E+04	2.88E+04	4.93E+03	4.86E+03	6.38E+03	1.82E+04
F10	STD	3.72E+02	6.60E+02	1.17E+03	1.76E+03	4.19E+02	4.47E+02	6.27E+02	9.45E+02	5.53E+02	5.31E+02	6.99E+02	1.13E+03
	RANK	2	3	**1**	**1**	4	4	4	4	3	2	2	3
	AVG	**1.11E+03**	1.24E+03	**1.27E+03**	4.29E+03	1.20E+03	**1.21E+03**	1.45E+03	**3.04E+03**	2.13E+03	2.05E+03	5.87E+03	6.95E+04
F11	STD	**6.73E+00**	6.71E+01	**3.95E+01**	6.84E+02	2.83E+01	**3.24E+01**	1.10E+02	**2.95E+02**	5.99E+02	4.12E+02	1.42E+03	1.35E+04
	RANK	**1**	2	**1**	2	3	**1**	3	**1**	5	5	5	5
	AVG	**1.21E+04**	**2.67E+05**	**3.01E+06**	**4.06E+07**	1.52E+09	1.01E+09	3.92E+10	2.56E+11	1.96E+08	2.13E+08	4.03E+09	2.54E+10
F12	STD	**1.04E+04**	**4.86E+05**	**2.50E+06**	**1.34E+07**	2.91E+09	8.45E+08	2.31E+10	5.42E+10	1.79E+08	2.47E+08	2.84E+09	1.12E+10
	RANK	**1**	**1**	**1**	**1**	6	5	5	5	5	4	4	4
	AVG	1.46E+04	**2.65E+04**	**1.14E+04**	**1.43E+04**	6.32E+08	1.16E+09	1.28E+10	4.58E+10	3.10E+08	2.07E+08	2.78E+09	1.81E+09
F13	STD	1.24E+04	**2.98E+04**	**7.50E+03**	**5.87E+03**	1.68E+09	2.99E+09	1.22E+10	1.68E+10	3.16E+08	2.35E+08	3.21E+09	1.31E+09
	RANK	2	**1**	**1**	**1**	5	5	5	5	4	4	4	4
	AVG	1.53E+03	1.59E+04	8.71E+04	6.08E+05	1.53E+03	**1.53E+03**	**1.79E+03**	**1.86E+05**	8.69E+05	1.06E+06	5.24E+06	1.65E+07
F14	STD	5.18E+01	1.24E+04	6.18E+04	**1.70E+05**	**2.52E+01**	**2.11E+01**	**5.64E+01**	2.24E+05	6.14E+05	6.04E+05	3.30E+06	6.20E+06
	RANK	3	3	2	2	2	**1**	**1**	**1**	5	5	5	5
	AVG	**1.53E+03**	5.61E+03	2.09E+04	**5.33E+03**	9.51E+03	**4.31E+03**	7.23E+08	9.21E+09	1.06E+07	5.61E+06	5.88E+08	7.86E+08
F15	STD	**5.18E+01**	**4.86E+03**	1.18E+04	**2.89E+03**	4.09E+04	1.16E+04	1.53E+09	7.91E+09	1.41E+07	6.73E+06	5.22E+08	8.92E+08
	RANK	**1**	2	2	**1**	3	**1**	5	5	5	5	4	4
	AVG	2.63E+03	2.68E+03	**2.56E+03**	**6.01E+03**	2.34E+03	**2.36E+03**	2.90E+03	7.14E+03	1.06E+07	3.14E+03	3.32E+03	6.96E+03
F16	STD	4.02E+02	2.59E+02	4.65E+02	7.52E+02	1.99E+02	**2.14E+02**	**4.20E+02**	9.97E+02	1.41E+07	2.80E+02	4.33E+02	8.44E+02
	RANK	3	2	**1**	**1**	2	**1**	2	4	6	5	3	3
	AVG	1.75E+03	2.18E+03	3.35E+03	**4.71E+03**	**1.72E+03**	**2.05E+03**	**2.70E+03**	5.42E+03	1.75E+03	1.78E+06	3.58E+03	7.27E+03
F17	STD	1.07E+01	2.06E+02	3.43E+02	**5.21E+02**	1.36E+01	**7.50E+01**	**2.59E+02**	1.10E+03	1.63E+01	1.42E+06	5.05E+02	1.34E+03
	RANK	3	2	3	**1**	**1**	**1**	**1**	3	4	5	5	5
	AVG	7.54E+03	1.63E+05	4.09E+05	1.17E+06	**2.07E+03**	5.46E+04	**2.95E+04**	**4.04E+05**	1.81E+06	1.60E+07	9.21E+06	1.44E+07
F18	STD	5.83E+03	1.35E+05	2.97E+05	3.79E+05	**9.39E+01**	1.98E+05	**2.76E+04**	**3.24E+05**	1.36E+06	2.31E+07	6.38E+06	6.74E+06
	RANK	2	3	2	2	**1**	2	**1**	**1**	5	5	5	4
	AVG	1.44E+04	5.34E+03	**2.73E+04**	**5.35E+03**	9.26E+04	**2.40E+03**	6.40E+07	9.38E+09	1.09E+07	2.45E+03	9.99E+07	1.09E+09
F19	STD	1.05E+04	3.19E+03	**1.07E+04**	**3.15E+03**	2.74E+05	1.98E+01	1.32E+08	6.89E+09	1.46E+07	2.95E+01	2.12E+08	1.53E+09
	RANK	2	6	**1**	**1**	3	**1**	4	5	5	4	5	4
	AVG	**2.00E+03**	**2.18E+03**	**2.60E+03**	5.52E+03	2.02E+03	2.22E+03	3.04E+03	5.94E+03	2.08E+03	2.43E+03	3.18E+03	**5.52E+03**
F20	STD	5.57E+01	**2.06E+02**	2.91E+02	4.99E+02	**1.21E+01**	5.50E+01	**1.37E+02**	3.17E+02	5.73E+01	2.75E+02	2.81E+02	4.67E+02
	RANK	**1**	**1**	**1**	2	2	3	3	3	4	5	4	**1**
	AVG	2.29E+03	2.45E+03	**2.54E+03**	3.25E+03	2.39E+03	2.86E+03	2.60E+03	**2.98E+03**	2.45E+03	**2.44E+03**	2.64E+03	3.28E+03
F21	STD	5.36E+01	**3.85E+01**	6.74E+01	1.79E+02	2.75E+01	1.07E+03	**6.07E+01**	1.58E+02	2.96E+01	5.10E+01	8.25E+01	**9.97E+01**
	RANK	2	2	**1**	2	4	3	2	**1**	5	**1**	3	4
	AVG	**2.30E+03**	3.95E+03	**7.79E+03**	2.00E+04	2.79E+03	**2.73E+03**	1.25E+04	3.09E+04	2.55E+03	2.84E+03	9.74E+03	2.18E+04
F22	STD	**1.75E+01**	2.35E+03	2.44E+03	1.44E+03	9.91E+02	**2.15E+01**	3.69E+03	9.75E+02	6.43E+02	4.01E+01	1.97E+03	1.00E+03
	RANK	**1**	6	**1**	2	4	**1**	4	4	3	3	3	3
	AVG	**2.62E+03**	2.54E+03	**2.98E+03**	3.70E+03	2.75E+03	**2.22E+03**	2.98E+03	3.64E+03	2.85E+03	2.48E+03	3.13E+03	**3.56E+03**
F23	STD	**7.74E+00**	1.76E+02	6.80E+01	1.72E+02	2.27E+01	**5.86E+01**	**4.79E+01**	**6.42E+01**	4.45E+01	1.40E+02	7.72E+01	8.29E+01
	RANK	**1**	5	**1**	3	3	**1**	2	2	5	3	4	**1**
	AVG	**2.75E+03**	**2.88E+03**	**3.14E+03**	4.54E+03	2.91E+03	2.92E+03	3.16E+03	4.82E+03	3.11E+03	3.12E+03	3.51E+03	**4.36E+03**
F24	STD	**5.89E+00**	6.16E+01	9.11E+01	**1.23E+02**	2.60E+01	**2.82E+01**	5.87E+01	1.82E+02	6.10E+01	8.65E+01	1.38E+02	1.47E+02
	RANK	**1**	**1**	**1**	3	3	2	2	4	5	5	4	**1**
	AVG	2.93E+03	**2.90E+03**	**3.08E+03**	**3.41E+03**	3.14E+03	3.16E+03	5.44E+03	1.15E+04	2.98E+03	2.98E+03	3.45E+03	4.86E+03
F25	STD	2.06E+01	**2.05E+01**	**2.98E+01**	**4.80E+01**	6.55E+01	7.33E+01	5.84E+02	1.28E+03	3.79E+01	2.96E+01	1.50E+02	3.44E+02
	RANK	2	**1**	**1**	**1**	5	5	5	5	4	4	4	3
	AVG	**2.90E+03**	**4.10E+03**	**2.90E+03**	2.06E+04	5.50E+03	5.26E+03	8.50E+03	2.41E+04	4.41E+03	4.32E+03	5.32E+03	1.72E+04
F26	STD	**1.25E+02**	1.57E+03	3.52E+03	8.35E+03	6.18E+02	**4.93E+02**	8.58E+02	2.05E+03	1.11E+03	1.09E+03	1.58E+03	5.39E+03
	RANK	**1**	**1**	**1**	4	5	4	5	5	3	2	2	2
	AVG	**3.09E+03**	**3.14E+03**	**3.47E+03**	**3.81E+03**	3.26E+03	3.26E+03	3.67E+03	4.62E+03	3.26E+03	3.26E+03	3.65E+03	3.82E+03
F27	STD	**2.16E+00**	2.32E+01	1.69E+02	2.32E+02	1.87E+01	2.52E+01	1.42E+02	**1.11E+02**	1.91E+01	1.57E+01	1.30E+02	1.16E+02
	RANK	**1**	**1**	**1**	**1**	4	3	5	5	5	4	4	2
	AVG	**3.17E+03**	**3.13E+03**	**3.33E+03**	**3.51E+03**	3.87E+03	3.94E+03	6.06E+03	1.55E+04	3.41E+03	3.41E+03	3.87E+03	5.87E+03
F28	STD	4.58E+01	**3.82E+01**	**4.16E+01**	**3.26E+01**	2.02E+02	2.34E+02	3.70E+02	1.74E+03	6.31E+01	5.50E+01	2.11E+02	7.64E+02
	RANK	**1**	**1**	**1**	**1**	5	5	5	5	4	4	4	3
	AVG	**3.12E+03**	**3.38E+03**	**4.07E+03**	**7.49E+03**	3.16E+03	3.60E+03	4.18E+03	9.76E+03	3.26E+03	4.06E+03	4.74E+03	7.89E+03
F29	STD	5.81E+01	3.00E+02	3.91E+02	**2.61E+02**	8.31E+00	**9.64E+01**	**2.06E+02**	1.61E+03	5.35E+01	2.33E+02	4.05E+02	6.78E+02
	Rank	**1**	**1**	**1**	**1**	2	2	2	4	5	3	5	2
	AVG	**6.51E+03**	**1.33E+04**	**1.37E+06**	**1.12E+05**	1.39E+05	3.82E+06	4.03E+08	4.24E+10	1.18E+07	1.56E+07	1.58E+08	9.15E+08
F30	STD	**3.64E+03**	**1.04E+04**	**7.44E+05**	**5.09E+04**	2.23E+07	8.77E+06	6.90E+08	1.22E+10	1.12E+07	3.37E+07	3.93E+08	5.64E+08
	RANK	**1**	**1**	**1**	**1**	2	3	5	5	6	5	3	3
Sum of ranks		**42**	**66**	**50**	**53**	98	79	91	100	133	116	112	92
Average rank		**1.40**	**2.20**	**1.67**	**1.86**	3.27	2.63	3.03	3.45	4.43	3.87	3.73	3.14
Overall rank		**1**	**1**	**1**	**1**	3	2	3	4	5	5	5	3

TABLE 4. *(Continued.)* **Comparison results of DECEHGS and other advanced algorithms on CEC2017 test suites with different dimensions.**

Function		mGWO				HIWOA				PPSO			
Dim		*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100	*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100	*Dim*=10	*Dim*=30	*Dim*=50	*Dim*=100
F1	AVG	1.89E+06	4.32E+11	9.20E+09	1.05E+11	1.05E+11	1.05E+09	9.14E+11	2.55E+12	1.03E+04	4.59E+08	2.32E+09	1.91E+07
	STD	4.95E+08	5.33E+08	4.01E+09	6.35E+10	3.48E+10	2.43E+10	6.59E+10	1.02E+11	9.12E+03	2.47E+09	1.02E+10	6.64E+09
	RANK	3	6	3	3	6	3	6	6	2	2	2	2
F3	AVG	4.04E+04	8.39E+04	1.20E+05	3.41E+05	8.50E+04	4.13E+04	1.80E+05	3.53E+05	6.24E+02	4.01E+03	6.62E+04	4.62E+05
	STD	8.04E+03	8.03E+03	1.84E+04	2.31E+04	4.65E+03	5.49E+03	1.67E+04	**9.36E+03**	5.10E+02	5.27E+03	2.25E+04	1.01E+05
	RANK	4	6	4	4	6	4	6	5	3	3	3	6
F4	AVG	4.08E+02	8.56E+03	7.89E+02	2.23E+03	5.70E+02	5.37E+02	2.60E+04	8.18E+04	4.15E+02	5.05E+02	5.73E+02	8.82E+02
	STD	2.17E+01	2.97E+01	8.09E+01	7.23E+02	9.87E+02	8.70E+02	4.13E+03	7.13E+03	2.32E+01	3.15E+01	5.81E+01	1.01E+02
	RANK	2	6	3	3	5	3	6	6	3	2	2	2
F5	AVG	6.50E+02	9.26E+02	8.26E+02	1.43E+03	9.22E+02	6.46E+02	1.18E+03	2.08E+03	5.26E+02	6.58E+02	8.33E+02	1.38E+03
	STD	4.58E+01	4.27E+01	6.38E+01	1.13E+02	1.77E+01	**1.34E+01**	**2.42E+01**	**2.58E+01**	1.13E+01	4.12E+01	8.05E+01	9.26E+01
	RANK	4	6	3	5	6	2	6	6	2	3	4	4
F6	AVG	6.14E+02	7.03E+02	**6.35E+02**	**6.64E+02**	7.02E+02	**6.17E+02**	7.38E+02	7.25E+02	6.18E+02	6.58E+02	6.81E+02	6.82E+02
	STD	4.60E+00	**4.38E+00**	8.90E+00	8.95E+00	5.16E+00	5.47E+00	**5.72E+00**	**2.20E+00**	8.74E+00	1.57E+01	8.27E+00	7.30E+00
	RANK	2	6	**1**	**1**	6	**1**	6	6	4	3	3	4
F7	AVG	9.50E+02	1.37E+03	1.27E+03	2.43E+03	1.38E+03	9.45E+02	1.93E+03	3.90E+03	**7.50E+02**	9.48E+02	**1.19E+03**	2.55E+03
	STD	3.26E+01	3.84E+01	5.51E+01	1.48E+02	2.66E+01	2.81E+01	**4.11E+01**	6.83E+01	**1.66E+01**	5.18E+01	9.16E+01	2.39E+02
	RANK	4	6	4	2	6	2	6	6	**1**	3	**1**	4
F8	AVG	9.44E+02	1.14E+03	1.14E+03	1.70E+03	1.14E+03	9.37E+02	1.49E+03	2.56E+03	8.36E+02	1.05E+03	1.37E+03	2.34E+03
	STD	3.97E+01	4.01E+01	7.35E+01	1.14E+02	1.19E+01	**1.66E+01**	**2.37E+01**	3.14E+01	1.09E+01	4.44E+01	7.59E+01	1.24E+02
	RANK	5	6	3	3	6	2	6	6	2	5	5	5
F9	AVG	1.34E+03	1.17E+04	**5.52E+03**	3.42E+04	1.14E+04	**1.37E+03**	4.02E+04	8.20E+04	1.01E+03	3.97E+03	1.26E+04	3.20E+04
	STD	3.98E+02	**3.64E+02**	1.94E+03	7.89E+03	1.05E+03	9.87E+02	2.34E+03	3.31E+03	1.11E+02	1.30E+03	2.77E+03	5.10E+03
	RANK	3	6	**1**	3	6	**1**	6	6	2	3	4	2
F10	AVG	8.25E+03	8.38E+03	1.53E+04	3.16E+04	8.43E+03	8.38E+03	1.52E+04	3.15E+04	**1.69E+03**	**4.61E+03**	7.47E+03	1.59E+04
	STD	2.97E+02	3.22E+02	8.09E+02	**5.87E+02**	**2.45E+02**	**3.01E+02**	**5.21E+02**	7.92E+02	2.75E+02	5.50E+02	1.11E+03	1.59E+03
	RANK	5	5	6	6	6	6	5	5	**1**	**1**	3	2
F11	AVG	1.39E+03	7.58E+03	2.33E+03	6.88E+04	7.52E+03	1.37E+03	2.10E+04	2.10E+05	1.13E+03	1.27E+03	1.44E+03	1.01E+04
	STD	5.24E+01	4.95E+01	3.17E+02	1.07E+04	1.35E+03	1.41E+03	1.70E+03	2.14E+04	2.83E+01	6.09E+01	2.01E+02	1.69E+04
	RANK	4	6	4	4	6	4	6	6	2	3	2	3
F12	AVG	1.80E+08	8.93E+10	1.86E+09	1.89E+10	1.18E+08	1.24E+08	5.78E+11	1.52E+12	9.10E+05	1.78E+06	3.05E+08	1.22E+09
	STD	1.10E+08	5.72E+07	1.58E+09	7.54E+09	1.17E+10	1.06E+10	7.00E+10	1.10E+11	2.46E+06	2.97E+06	1.53E+09	4.47E+09
	RANK	4	6	3	3	3	3	6	6	2	2	2	2
F13	AVG	7.88E+06	5.57E+10	8.72E+07	5.74E+08	5.69E+10	6.54E+06	3.62E+11	3.83E+11	**7.87E+03**	3.14E+04	6.29E+08	5.53E+05
	STD	1.63E+07	7.07E+06	5.27E+07	3.36E+08	1.45E+10	1.32E+10	5.93E+10	2.89E+10	**6.07E+03**	3.02E+04	1.80E+09	2.84E+06
	RANK	3	6	2	3	6	3	6	6	**1**	2	3	2
F14	AVG	9.54E+04	3.48E+06	8.87E+05	1.34E+07	3.15E+06	1.03E+05	4.56E+07	5.01E+07	**1.49E+03**	1.26E+04	2.36E+05	1.10E+06
	STD	5.79E+04	7.67E+04	7.72E+05	7.77E+06	1.29E+06	1.48E+06	1.73E+07	8.92E+06	5.86E+01	3.59E+04	4.44E+05	1.03E+06
	RANK	4	6	4	4	6	4	6	6	**1**	2	3	3
F15	AVG	2.34E+05	2.37E+09	8.39E+06	8.74E+07	2.41E+09	1.77E+05	4.05E+10	1.94E+11	1.66E+03	8.55E+03	**1.31E+04**	2.76E+08
	STD	2.26E+05	1.59E+05	9.74E+06	5.37E+07	7.75E+08	8.46E+08	6.77E+09	1.23E+10	1.74E+02	8.54E+03	**8.39E+03**	1.49E+09
	RANK	4	6	3	2	6	4	6	6	2	3	**1**	3
F16	AVG	2.95E+03	5.19E+03	3.40E+03	9.00E+03	5.10E+03	2.92E+03	7.51E+03	2.14E+04	**1.77E+03**	2.78E+03	3.86E+03	6.54E+03
	STD	4.12E+02	3.60E+02	7.32E+02	1.27E+03	3.00E+02	2.28E+02	4.28E+02	1.02E+03	**1.24E+02**	3.30E+02	4.31E+02	**7.15E+02**
	RANK	4	6	4	5	5	4	6	6	**1**	3	5	2
F17	AVG	1.75E+03	2.24E+07	3.89E+03	7.19E+03	1.83E+03	1.68E+06	3.50E+03	1.02E+06	1.74E+03	1.09E+05	2.71E+03	4.99E+03
	STD	1.03E+01	1.43E+06	3.51E+02	9.09E+02	**1.48E+01**	1.02E+07	4.13E+02	8.10E+05	1.65E+01	9.45E+04	2.49E+02	6.25E+02
	RANK	5	6	6	4	6	4	4	6	2	3	2	2
F18	AVG	1.14E+06	2.24E+09	4.45E+06	1.45E+07	2.36E+07	8.21E+05	1.60E+08	1.10E+08	1.20E+04	**2.12E+04**	7.25E+05	2.26E+06
	STD	9.06E+05	9.37E+05	2.64E+06	9.14E+06	9.56E+06	6.35E+08	4.45E+07	4.09E+07	1.19E+04	**3.15E+04**	1.74E+06	1.43E+06
	RANK	4	6	4	5	6	4	6	6	3	**1**	3	3
F19	AVG	5.00E+05	2.71E+03	9.20E+06	1.40E+08	2.11E+09	2.45E+03	2.39E+10	2.01E+11	**2.27E+03**	2.45E+03	7.51E+04	1.16E+05
	STD	5.60E+05	4.26E+01	9.67E+06	7.40E+07	7.09E+08	**1.80E+01**	4.51E+09	2.75E+10	**9.82E+02**	3.56E+01	1.97E+05	4.49E+05
	RANK	4	5	3	3	6	2	6	6	**1**	3	2	2
F20	AVG	2.45E+03	2.18E+03	3.86E+03	7.26E+03	2.24E+03	3.75E+03	3.24E+03	7.66E+03	2.06E+03	2.34E+03	2.93E+03	6.22E+03
	STD	4.31E+01	2.66E+02	3.46E+02	3.32E+02	2.73E+01	3.01E+02	2.82E+02	3.02E+02	5.57E+01	1.73E+02	4.21E+02	**2.57E+02**
	RANK	6	2	6	5	5	6	5	6	3	4	2	4
F21	AVG	**2.05E+03**	8.15E+03	2.65E+03	3.28E+03	2.71E+03	3.19E+03	3.11E+03	4.62E+03	2.33E+03	5.03E+03	2.66E+03	3.29E+03
	STD	**1.13E+01**	2.24E+03	7.94E+01	1.21E+02	1.45E+01	5.44E+02	**3.05E+01**	1.10E+02	2.69E+01	1.87E+03	8.78E+01	1.39E+02
	RANK	**1**	6	4	3	6	4	6	6	3	5	5	5
F22	AVG	3.74E+03	3.41E+03	1.63E+04	3.42E+04	8.06E+03	2.82E+03	1.72E+04	3.48E+04	2.32E+03	2.88E+03	9.22E+03	**1.87E+04**
	STD	2.73E+03	4.33E+01	4.85E+02	7.34E+02	5.02E+02	7.51E+01	**4.01E+02**	5.45E+02	9.90E+01	6.79E+01	**7.67E+02**	1.48E+03
	RANK	5	5	5	5	6	2	6	6	2	4	2	**1**
F23	AVG	2.82E+03	2.51E+03	3.11E+03	3.77E+03	3.41E+03	2.90E+03	4.24E+03	6.33E+03	2.63E+03	2.42E+03	3.28E+03	4.08E+03
	STD	4.76E+01	2.32E+02	6.64E+01	1.32E+02	6.24E+01	1.07E+02	8.24E+01	2.02E+02	1.21E+01	1.43E+02	1.46E+02	2.15E+02
	RANK	4	4	3	4	6	6	6	6	2	2	5	5
F24	AVG	2.99E+03	3.62E+03	3.25E+03	4.38E+03	3.60E+03	3.00E+03	4.55E+03	9.98E+03	2.75E+03	3.10E+03	3.52E+03	5.27E+03
	STD	4.82E+01	4.33E+01	**5.74E+01**	1.28E+02	9.37E+01	7.55E+01	1.70E+02	5.28E+02	4.80E+01	9.06E+01	1.90E+02	3.98E+02
	RANK	4	6	3	2	6	3	6	6	2	4	5	5
F25	AVG	**2.93E+03**	4.30E+03	3.22E+03	5.10E+03	4.25E+03	2.92E+03	1.13E+04	2.31E+04	2.94E+03	2.91E+03	3.09E+03	3.50E+03
	STD	**1.69E+01**	1.59E+01	7.34E+01	7.06E+02	1.57E+02	1.67E+02	1.05E+03	1.92E+03	2.60E+01	2.69E+01	8.44E+01	6.11E+01
	RANK	**1**	6	3	4	6	3	6	6	3	2	2	2
F26	AVG	5.22E+03	1.01E+04	7.32E+03	**1.65E+04**	1.02E+04	5.25E+03	1.58E+04	4.60E+04	3.09E+03	5.35E+03	8.26E+03	1.85E+04
	STD	4.10E+02	3.87E+02	7.42E+02	**1.25E+03**	3.72E+02	4.21E+02	**3.22E+02**	1.92E+03	3.82E+02	1.38E+03	1.48E+03	2.58E+03
	RANK	4	6	3	**1**	6	3	6	6	2	5	4	3
F27	AVG	3.24E+03	4.13E+03	3.60E+03	4.11E+03	4.08E+03	3.24E+03	6.62E+03	1.30E+04	3.11E+03	3.26E+03	3.60E+03	3.92E+03
	STD	1.23E+01	**1.28E+01**	**7.64E+01**	1.61E+02	1.75E+02	1.35E+02	4.64E+02	7.79E+02	2.46E+01	4.86E+01	1.57E+02	2.16E+02
	RANK	3	6	2	4	6	2	6	6	2	5	3	3
F28	AVG	3.32E+03	5.85E+03	3.72E+03	7.25E+03	5.86E+03	3.35E+03	1.12E+04	2.61E+04	3.29E+03	3.26E+03	3.51E+03	3.80E+03
	STD	**3.27E+01**	5.76E+01	3.07E+02	1.88E+03	1.67E+02	2.13E+02	1.03E+03	1.72E+03	1.26E+02	6.14E+01	8.36E+02	3.93E+02
	RANK	3	6	3	4	6	3	6	6	2	2	2	2
F29	AVG	3.19E+03	4.07E+03	5.32E+03	1.11E+04	3.40E+03	6.39E+03	4.64E+03	2.28E+05	3.21E+03	5.82E+03	4.24E+03	7.91E+03
	STD	**1.81E+01**	2.86E+02	6.12E+02	8.78E+02	5.62E+01	3.90E+02	3.54E+02	9.11E+04	4.53E+01	4.25E+02	3.54E+02	3.45E+02
		3	4	6	5	6	6	4	6	4	5	3	3
F30	AVG	9.23E+06	1.10E+10	2.92E+08	1.45E+09	9.17E+06	1.12E+07	4.76E+10	3.55E+11	4.00E+05	3.90E+04	2.06E+06	5.89E+07
	STD	5.18E+06	9.05E+06	1.41E+08	5.65E+08	3.96E+09	3.54E+09	1.07E+10	2.13E+10	5.41E+05	7.62E+04	1.11E+06	3.08E+08
	RANK	5	6	4	4	4	4	6	6	3	2	2	2
Sum of ranks		107	163	103	104	166	98	168	172	63	87	85	88
Average rank		3.57	5.43	3.43	3.59	5.53	3.27	5.60	5.93	2.10	2.90	2.83	3.03
Overall rank		4	6	4	5	6	4	6	6	2	3	2	2

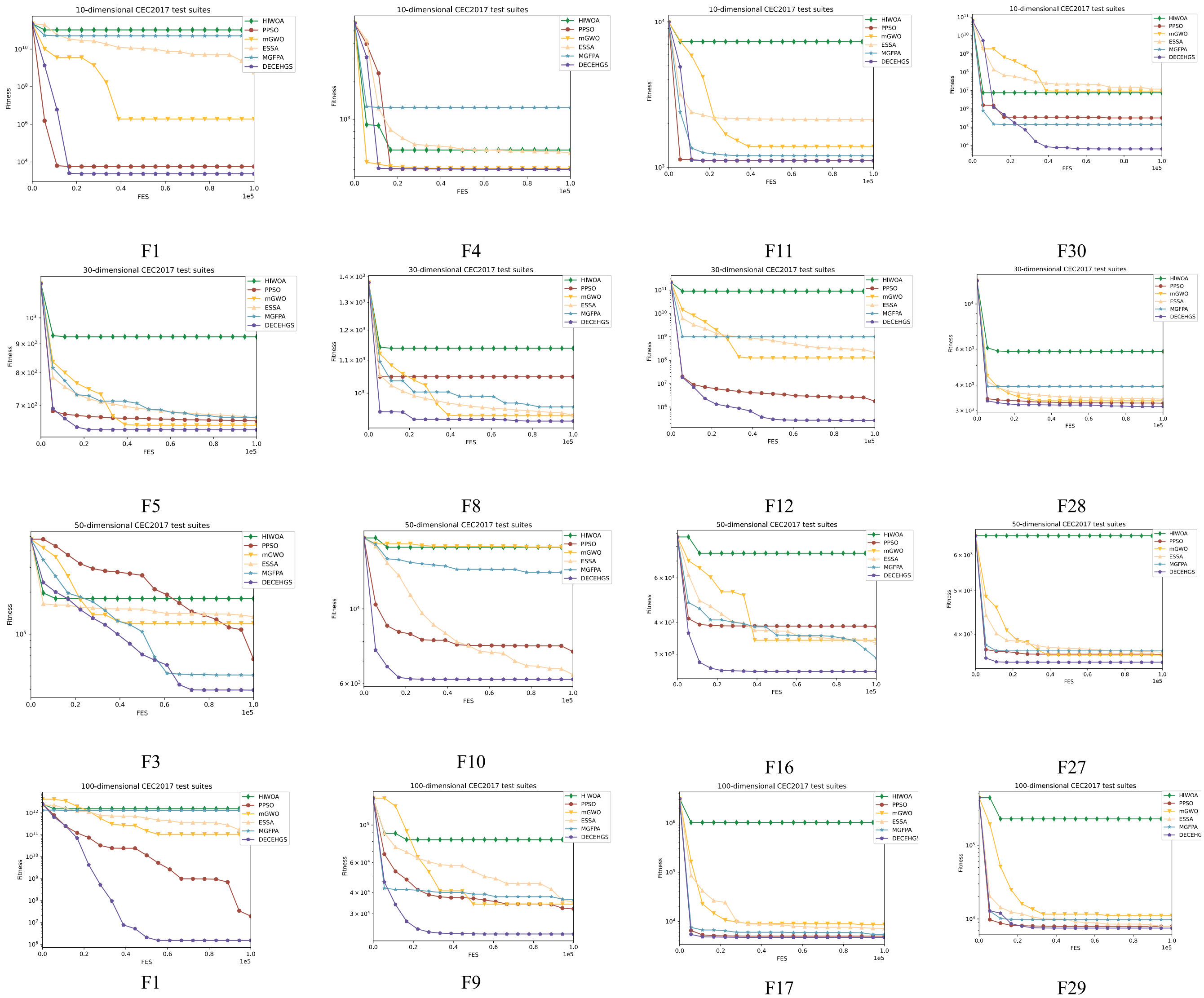


FIGURE 4. Convergence trends curves for DECEHGS versus other advanced algorithms on CEC2017 test suites with different dimensions.

TABLE 5. Comparison results of DECEHGS and other methods for WBD problem.

Algorithm	H	L	t	b	$cost$
DECEHGS	0.205212	3.470466	9.036321	0.21524	1.724745
SADE	0.205212	3.470521	9.036787	0.20595	1.724813
SHADE	0.205736	3.470545	9.036641	0.20574	1.724813
LSHcEpS	0.203725	3.515719	9.042718	0.20571	1.724822
MPADE	0.205696	3.471308	9.036594	0.20573	1.724852
HGS	0.205535	3.474330	9.037526	0.20573	1.725190
GWO	0.205615	3.470297	9.047163	0.20568	1.726081
MVO	0.203262	3.528347	9.030419	0.20605	1.730163
SCA	0.219027	3.412614	9.129591	0.32146	1.732615
HHO	0.209225	3.372327	9.103551	0.20944	1.756623
GA	0.219160	3.261167	8.905067	0.21960	1.797007
BBO	0.136604	6.373279	8.574036	0.23366	2.095000
DAVID	0.221628	0.231434	0.253285	0.22342	2.134113
HS	0.253187	0.218421	0.249365	0.26220	2.156022

TABLE 6. Comparison results of DECEHGS and other methods for TCSD problem.

Algorithm	d	D	N	$cost$
DECEHGS	0.0520	0.3644	10.8489	0.012664
MPADE	0.0550	0.3567	11.2886	0.012664
MFO	0.0522	0.3687	10.6180	0.012667
HGS	0.0512	0.3448	12.0250	0.012668
SHADE	0.0517	0.3567	11.2890	0.012688
GSA	0.0500	0.3168	14.1060	0.012744
MVO	0.0500	0.3167	14.1314	0.012773
GA	0.0503	0.3212	13.9820	0.013000
LSHcEpS	0.0524	0.3567	11.2876	0.013321
BAA	0.0498	0.3162	14.1392	0.014218
BBO	0.0601	0.5553	5.61491	0.015273

3) PRESSURE VESSEL DESIGN (PVD) PROBLEM

The third engineering design problem aims to solve the cylindrical pressure vessel and determine the optimal values of the depth of the shell (T_s), head (T_h), the internal radius (R), and the extent of the section, minus the head (L). The proposed algorithm was compared with referenced methods such as LSHcEpS, SADE, ES, GA, PSO, GSA, HGS, JAYA [75] and MPADE methods. The mathematical model

TABLE 7. Comparison results of DECEHGS and other methods for PVD problem.

Algorithm	T_h	T_s	R	L	cost
DECEHGS	0.774531	0.383204	40.31962	198.9731	5870.1238
LSHcEpS	0.774552	0.383204	40.31965	199.9996	5870.1239
HGS	0.846705	0.454419	50.23903	196.3444	5875.2534
SADE	0.778629	0.385189	40.52936	197.1041	5877.1213
MPADE	0.803148	0.431720	42.22391	197.2276	5881.3548
PSO	0.921317	0.402357	42.22107	173.3536	5926.1723
JAYA	0.869483	0.321524	40.56362	197.2015	5937.8312
ES	0.873292	0.433235	45.62113	136.9033	6073.0212
GA	1.042158	0.575156	49.72151	103.2373	7073.7191
GSA	1.121268	0.734874	53.12723	119.3829	8708.5977

is stated as follows:

$$
\begin{aligned}
&consider:\\
&\quad x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]\\
&\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2\\
&\qquad\qquad + 3.1661x_1^2x_4 + 19.84x_1^2x_3\\
&subject\ to:\\
&\quad g_1(x) = -x_1 + 0.0193x_3 \le 0\\
&\quad g_2(x) = -x_2 + 0.00954x_3 \le 0\\
&\quad g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0\\
&\quad g_4(x) = x_4 - 240 \le 0
\end{aligned}
$$

In Table 7, it can be seen that the DECEHGS and LSHcEpS occupied the first and the second ranks for solving this function. The cylindrical PV obtains the lowest total cost when T_s, T_h, R, and L are set to be 0.774531, 0.383204, 40.31962, and 198.9731. Among all of these algorithms, DECEHGS can find a feasible optimal design.

4) THREE BAR TRUSS DESIGN (TBTD) PROBLEM

Three bar truss design is a structural optimization problem in the field of civil engineering. Two parameters x_1 and x_2 are manipulated to minimize the weight subject to stress, deflection, and buckling constraints. This problem can be formulated as follows:

$$
\begin{aligned}
&consider:\\
&\quad x = [x_1, x_2]\\
&\min f(x) = (2\sqrt{2}x_1 + x_2) \times l\\
&\quad g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}p - \sigma \le 0\\
&\quad g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}p - \sigma \le 0\\
&\quad g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1}p - \sigma \le 0\\
&variablesrange:\\
&\quad 0 \le x_i \le 1, i = 1, 2\\
&\quad l = 100cm, p = 2kN/cm^2, \sigma = 2kN/cm^2
\end{aligned}
$$

The MPEDE, SADE, cuckoo search algorithm (CS) [76], GA, GWO, MFO, HGS, WOA and MPADE have been investigated for this case. Table 8 summarizes the obtained results by the DECEHGS and the other algorithms. It can be seen that the DECEHGS, MPEDE, and SADE achieved the best results. HGS obtained the fourth-best design while other algorithms converged to higher costs than the top three algorithms. The CS algorithm has an inferior rank to those methods.

TABLE 8. Comparison results of DECEHGS and other methods for TBTD problem.

Algorithm	x_1	x_2	cost
DECEHGS	0.788675	0.408254	263.8957998
MPEDE	0.788675	0.408248	263.8957998
SADE	0.788675	0.408248	263.8957998
HGS	0.788687	0.408211	263.8957999
MPADE	0.788675	0.408248	263.8957999
LSHcEpS	0.788677	0.408252	263.8958433
WOA	0.788531	0.408245	263.8959538
MFO	0.788220	0.409535	263.8959541
GWO	0.788978	0.407391	263.8961268
GA	0.787462	0.411744	263.9033177
CS	0.779558	0.450004	265.2926725

VI. CONCLUSION

Hunger games search algorithm is a simple and efficient algorithm but tends to stagnate in local optimal and remains premature convergence. In this paper, we enhance the non-cooperative individuals by using the operators of differential evolution. For the traditional evolutionary population dynamics strategy, we propose a modified EPD technique to increase the diversity of the population, improve the quality of the solution and avoid local optimal. Furthermore, a chaotic local search strategy is introduced to prevent premature convergence. However, DECEHGS also has its limitations. With the improvement of computing performance, the time complexity is correspondingly higher.

In the end, we use three groups of experiments to verify the performance of DECEHGS. IEEE CEC2014 and IEEE CEC2017 test suites were chosen to compare DECEHGS to the state-of-the-art algorithms, including ILSHADE, LSHADE_cnEpSi, JSO, CoDE, BBPSO, ISCA, OBWOA, TVBSSA, ESSA, MGFPA, mGWO, HIWOA, and PPSO. The experimental results indicate that the DECEHGS algorithm outperforms other methods. In addition, four real-world engineering problems were used to verify the performance of DECEHGS. The comparison results reveal that the performance of DECEHGS is better than many other methods.

In future works, DECEHGS can be designed as distributed or parallel algorithms to solve large-scale problems such as cloud workflow scheduling problems, power electronic circuit problems, and airline crew rostering problems. Moreover, it is interesting to extend DECEHGS to a multi-objective version to solve multi-objective optimization problems, including job shop scheduling, supply chain configuration, and vehicle dispatch problems.

ACKNOWLEDGMENT

The authors would like to thank Dr. Huiling Chen for his helpful suggestions and anonymous reviewers' valuable comments.

REFERENCES

[1] J.-Y. Li, Z.-H. Zhan, J. Xu, S. Kwong, and J. Zhang, "Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 20, 2021, doi: 10.1109/TNNLS.2021.3106399.

[2] X.-F. Liu, Z.-H. Zhan, T.-L. Gu, S. Kwong, Z. Lu, H. B.-L. Duh, and J. Zhang, "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1557–1570, May 2020, doi: 10.1109/TNNLS.2019.2920887.

[3] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106367.

[4] S. E. Shukri, R. Al-Sayyed, A. Hudaib, and S. Mirjalili, "Enhanced multi-verse optimizer for task scheduling in cloud computing environments," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114230.

[5] M. A. Elaziz, S. Xiong, K. P. N. Jayasena, and L. Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution," *Knowl.-Based Syst.*, vol. 169, pp. 39–52, Apr. 2019.

[6] R. Pitakaso, K. Sethanan, and T. Jamrus, "Hybrid PSO and ALNS algorithm for software and mobile application for transportation in ice manufacturing industry 3.5," *Comput. Ind. Eng.*, vol. 144, Jun. 2020, Art. no. 106461.

[7] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li, and S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, Dec. 2018.

[8] M. Alweshah, S. Alkhalaileh, D. Albashish, M. Mafarja, Q. Bsoul, and O. Dorgham, "A hybrid mine blast algorithm for feature selection problems," *Soft Comput.*, vol. 25, no. 1, pp. 517–534, Jan. 2021.

[9] H. A. Abdulwahab, A. Noraziah, A. A. Alsewari, and S. Q. Salih, "An enhanced version of black hole algorithm via levy flight for optimization and data clustering problems," *IEEE Access*, vol. 7, pp. 142085–142096, 2019.

[10] B. B. Gupta and M. Quamara, "An identity based access control and mutual authentication framework for distributed cloud computing services in IoT environment using smart cards," *Proc. Comput. Sci.*, vol. 132, pp. 189–197, 2018.

[11] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 1507, pp. 1–52, Jul. 2021, doi: 10.1007/s10462-021-10042-y.

[12] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, S. Mirjalili, Ed., Cham, Switzerland: Springer, 2019, pp. 43–55.

[13] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.

[14] Y. Zhang and X. Gu, "Biogeography-based optimization algorithm for large-scale multistage batch plant scheduling," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113776.

[15] A. Auger and N. Hansen, "Tutorial CMA-ES: Evolution strategies and covariance matrix adaptation," in *Proc. 14th Int. Conf. Genetic Evol. Comput. Conf. Companion (GECCO Companion)*, Philadelphia, PA, USA, 2012, pp. 827–848, doi: 10.1145/2330784.2330919.

[16] A. Ala'a, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Comprehensive review of the development of the harmony search algorithm and its applications," *IEEE Access*, vol. 7, pp. 14233–14245, 2019.

[17] S. Q. Salih and A. A. Alsewari, "A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 10359–10386, Jul. 2020.

[18] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Appl. Math. Model.*, vol. 71, pp. 45–59, Jul. 2019.

[19] J.-R. Jian, Z.-G. Chen, Z.-H. Zhan, and J. Zhang, "Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 779–793, Aug. 2021.

[20] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Jul. 2015.

[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[22] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm for optimization of truss structures with discrete variables," *Comput. Struct.*, vols. 102–103, pp. 49–63, Jul. 2012.

[23] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016, doi: 10.1007/s00521-015-1870-7.

[24] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113396.

[25] F. S. Gharehchopogh, H. Shayanfar, and H. Gholizadeh, "A comprehensive survey on symbiotic organisms search algorithms," *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 2265–2312, Mar. 2020, doi: 10.1007/s10462-019-09733-4.

[26] B. S. Yildiz, H. Lekesiz, and A. R. Yildiz, "Structural design of vehicle components using gravitational search and charged system search algorithms," *Mater. Test.*, vol. 58, no. 1, pp. 79–81, Jan. 2016, doi: 10.3139/120.110819.

[27] P. C. S. Rao, A. J. S. Kumar, Q. Niyaz, P. Sidike, and V. K. Devabhaktuni, "Binary chemical reaction optimization based feature selection techniques for machine learning classification problems," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114169.

[28] Y.-J. Zheng, X.-Q. Lu, Y.-C. Du, Y. Xue, and W.-G. Sheng, "Water wave optimization for combinatorial optimization: Design strategies and applications," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105611.

[29] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Aloufi, and K. Z. Zamli, "Hybrid harmony search algorithm with grey wolf optimizer and modified opposition-based learning," *IEEE Access*, vol. 7, pp. 68764–68785, 2019.

[30] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.

[31] S. M. Mahmoudi and M. Aghaie, "Evaluation of fuzzy based HS and GSA on reloading cycle length optimization of PWR nuclear power plant," *Ann. Nucl. Energy*, vol. 134, pp. 1–10, Dec. 2019.

[32] Q. Fan, W. Wang, and X. Yan, "Differential evolution algorithm with strategy adaptation and knowledge-based control parameters," *Artif. Intell. Rev.*, vol. 51, no. 2, pp. 219–253, Feb. 2019, doi: 10.1007/s10462-017-9562-6.

[33] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114864.

[34] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[35] M. Abdel-Basset, A.-N. Hessin, and L. Abdel-Fatah, "A comprehensive study of cuckoo-inspired algorithms," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 345–361, Jan. 2018, doi: 10.1007/s00521-016-2464-8.

[36] R. Nagal, P. Kumar, and P. Bansal, "Optimization of adaptive noise canceller with grey wolf optimizer for EEG/ERP signal noise cancellation," in *Proc. 6th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Mar. 2019, pp. 670–675.

[37] M. Pant, R. Thangaraj, C. Grosan, and A. Abraham, "Hybrid differential evolution–particle swarm optimization algorithm for solving global optimization problems," in *Proc. 3rd Int. Conf. Digit. Inf. Manage.*, Nov. 2008, pp. 18–24, doi: 10.1109/ICDIM.2008.4746766.

[38] G. Wu, R. Mallipeddi, P. N. Suganthanc, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Inf. Sci.*, vol. 329, pp. 329–345, Feb. 2016.

[39] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 390–401, Jan. 2013.

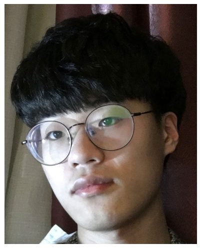
[40] J. M. Cushing, "Difference equations as models of evolutionary population dynamics," *J. Biol. Dyn.*, vol. 13, no. 1, pp. 103–127, Jan. 2019, doi: 10.1080/17513758.2019.1574034.

[41] H. Chen, "Global chaos synchronization of new chaotic systems via nonlinear control," *Chaos, Solitons Fractals*, vol. 23, no. 4, pp. 1245–1251, Feb. 2005.

[42] F. Mascia, M. López-Ibáñez, J. Dubois-Lacoste, M.-É. Marmion, and T. Stützle, "Algorithm comparison by automatically configurable stochastic local search frameworks: A case study using flow-shop scheduling problems," in *Hybrid Metaheuristics*, M. J. Blesa, C. Blum, S. Voß, Eds. Cham, Switzerland: Springer-Verlag, Jun. 2014, pp. 30–44.
[43] W. Wei, C. M. Li, and H. Zhang, "Switching among non-weighting, clause weighting, and variable weighting in local search for SAT," in *Principles and Practice of Constraint Programming*, Berlin, Germany, 2008, pp. 313–326.
[44] E. Goles, "Self-organized critically: An explanation of 1/f noise," *Phys. Rev. Lett.*, vol. 56, no. 1, pp. 75–90, 1992.
[45] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: An explanation of 1/f noise," *Phys. Rev. Lett.*, vol. 71, no. 4, pp. 364–374, 1987.
[46] J. Yong, F. He, H. Li, and W. Zhou, "A novel bat algorithm based on collaborative and dynamic learning of opposite population," in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2018, pp. 541–546, doi: 10.1109/CSCWD.2018.8464759.
[47] M.-R. Chen, X. Li, X. Zhang, and Y.-Z. Lu, "A novel particle swarm optimizer hybridized with extremal optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 367–373, Mar. 2010.
[48] L. S. Fichter, E. J. Pyle, and S. J. Whitmeyer, "Expanding evolutionary theory beyond darwinism with elaborating, self-organizing, and fractionating complex evolutionary systems," *J. Geosci. Educ.*, vol. 58, no. 2, pp. 58–64, Mar. 2010, doi: 10.5408/1.3534853.
[49] M. A. Elaziz, D. Yousri, and S. Mirjalili, "A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics," *Adv. Eng. Softw.*, vol. 154, Apr. 2021, Art. no. 102973.
[50] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A.-Z. Ala'M, and S. Mirjalili, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018.
[51] D. Jia, G. Zheng, and M. K. Khan, "An effective memetic differential evolution algorithm based on chaotic local search," *Inf. Sci.*, vol. 181, no. 15, pp. 3175–3187, Aug. 2011.
[52] J. Brest, M. S. Maučec, and B. Bošković, "IL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 1188–1195, doi: 10.1109/CEC.2016.7743922.
[53] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 372–379, doi: 10.1109/CEC.2017.7969336.
[54] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1311–1318, doi: 10.1109/CEC.2017.7969456.
[55] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Jan. 2011, doi: 10.1109/TEVC.2010.2087271.
[56] J. Kennedy, "Bare bones particle swarms," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Apr. 2003, pp. 80–87, doi: 10.1109/SIS.2003.1202251.
[57] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan, "Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism," *Neural Comput. Appl.*, vol. 28, no. 10, pp. 2947–2958, Oct. 2017, doi: 10.1007/s00521-017-2837-7.
[58] M. A. El Aziz and D. Oliva, "Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm," *Energy Convers. Manage.*, vol. 171, pp. 1843–1859, Sep. 2018.
[59] H. Faris, A. A. Heidari, A. M. Al-Zoubi, M. Mafarja, I. Aljarah, M. Eshtay, and S. Mirjalili, "Time-varying hierarchical chains of salps with random weight networks for feature selection," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112898.
[60] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced salp swarm algorithm: Application to variable speed wind generators," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 82–96, Apr. 2019.
[61] M. K. Y. Shambour, A. A. Abusnaina, and A. I. Alsalibi, "Modified global flower pollination algorithm and its application for optimization problems," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 11, no. 3, pp. 496–507, Sep. 2019, doi: 10.1007/s12539-018-0295-2.
[62] C. Tang, W. Sun, W. Wu, and M. Xue, "A hybrid improved whale optimization algorithm," in *Proc. IEEE 15th Int. Conf. Control Automat. (ICCA)*, Jul. 2019, pp. 362–367, doi: 10.1109/ICCA.2019.8900003.
[63] M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, and L. Li, "Phasor particle swarm optimization: A simple and efficient variant of PSO," *Soft Comput.*, vol. 23, no. 19, pp. 9701–9718, Oct. 2019, doi: 10.1007/s00500-018-3536-8.
[64] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.
[65] H. Ahmadvand and A. Habibi, "A new second-order approximation method for optimum design of structures," *Austral. J. Civil Eng.*, vol. 19, no. 1, pp. 72–97, Jan. 2021, doi: 10.1080/14488353.2020.1798039.
[66] C. Li, D. E. Bernal, K. C. Furman, M. A. Duran, and I. E. Grossmann, "Sample average approximation for stochastic nonconvex mixed integer nonlinear programming via outer-approximation," *Optim. Eng.*, vol. 22, no. 3, pp. 1245–1273, Sep. 2021, doi: 10.1007/s11081-020-09563-2.
[67] 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 Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[68] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2. Sep. 2005, pp. 1785–1791, doi: 10.1109/CEC.2005.1554904.
[69] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.
[70] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Comput. Oper. Res.*, vol. 67, pp. 155–173, Mar. 2016.
[71] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[72] K. M. Ragsdell and D. T. Phillips, "Optimal design of a class of welded structures using geometric programming," *J. Eng. Ind.*, vol. 98, no. 3, pp. 1021–1025, 1976, doi: 10.1115/1.3438995.
[73] A. D. Belegundu and J. S. Arora, "A study of mathematical programmingmethods for structural optimization. Part II: Numerical results," *Int. J. Numer. Methods Eng.*, vol. 21, no. 9, pp. 1601–1623, Sep. 1985, doi: 10.1002/nme.1620210905.
[74] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[75] H. M. Pandey, "Jaya a novel optimization algorithm: What, how and why?" in *Proc. 6th Int. Conf. - Cloud Syst. Big Data Eng. (Confluence)*, Jan. 2016, pp. 728–730, doi: 10.1109/CONFLUENCE.2016.7508215.
[76] M. Mareli and B. Twala, "An adaptive cuckoo search algorithm for optimisation," *Appl. Comput. Informat.*, vol. 14, no. 2, pp. 107–115, Jul. 2018.

SHAOLANG LI received the bachelor's degree from the Department of Computer Science and Technology, Xinzhi College, Zhejiang Normal University, China. He is currently a Graduate Student (majoring in electronic information) with Zhejiang Normal University. His research interests include data mining, machine learning, and evolutionary computation.

XIAOBO LI received the B.Sc. degree from Nankai University, China, the Master of Engineering (Research) degree from The University of Sydney, Australia, and the Ph.D. degree from Zhejiang University, China. He is currently a full-time Professor with the Department of Computer Science and Engineering, College of Mathematics and Computer Science, Zhejiang Normal University, China. His current research interests include the different aspects of bioinformatics, machine learning, and data mining. He is a Professional Committee Member of the China Computer Federation and the Chinese Association for Artificial Intelligence. He has published more than 40 articles in international journals. He was awarded the Wu Wenjun Artificial Intelligence Science and Technology Prize in 2016.



YUXIN ZHAO received the bachelor's degree from the Software College, Jiangxi University of Science and Technology, China. He is currently pursuing the master's degree (majoring in electronic information) with Zhejiang Normal University, China. He is committed to researching intelligent optimization algorithms and machine learning.

HUI CHEN is currently a Graduate Student (majoring in electronic information) with Zhejiang Normal University, China. Her current research interests include swarm intelligence optimization algorithm, machine learning, and feature selection and their application to practical problems.

JUNWEI DONG received the bachelor's degree in mathematics and applied mathematics from Zhejiang Normal University, China. He is currently a Graduate Student (majoring in computer science and technology) with Zhejiang Normal University, China. His current research areas include machine learning and intelligent optimization algorithm.