Engineering with Computers (2023) 39:1735–1769
https://doi.org/10.1007/s00366-021-01545-x

ORIGINAL ARTICLE

Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems

Hongliang Zhang[1] · Tong Liu[2] · Xiaojia Ye[3] · Ali Asghar Heidari[1] · Guoxi Liang[4] · Huiling Chen[1] · Zhifang Pan[5]

Received: 22 February 2021 / Accepted: 2 November 2021 / Published online: 10 January 2022

Abstract

There is a new nature-inspired algorithm called salp swarm algorithm (SSA), due to its simple framework, it has been widely used in many fields. But when handling some complicated optimization problems, especially the multimodal and high-dimensional optimization problems, SSA will probably have difficulties in convergence performance or dropping into the local optimum. To mitigate these problems, this paper presents a chaotic SSA with differential evolution (CDESSA). In the proposed framework, chaotic initialization and differential evolution are introduced to enrich the convergence speed and accuracy of SSA. Chaotic initialization is utilized to produce a better initial population aim at locating a better global optimal. At the same time, differential evolution is used to build up the search capability of each agent and improve the sense of balance of global search and intensification of SSA. These mechanisms collaborate to boost SSA in accelerating convergence activity. Finally, a series of experiments are carried out to test the performance of CDESSA. Firstly, IEEE CEC2014 competition fuctions are adopted to evaluate the ability of CDESSA in working out the real-parameter optimization problems. The proposed CDESSA is adopted to deal with feature selection (FS) problems, then five constrained engineering optimization problems are also adopted to evaluate the property of CDESSA in dealing with real engineering scenarios. Experimental results reveal that the proposed CDESSA method performs significantly better than the original SSA and other compared methods.

Keywords Salp swarm algorithm · Chaotic initialization · Global optimization · Feature selection · Engineering optimization problems

Guoxi Liang, Huiling Chen and Zhifang Pan authors contributed equally to this work.

✉ Guoxi Liang
guoxiliang2017@gmail.com

✉ Huiling Chen
chenhuiling.jlu@gmail.com

✉ Zhifang Pan
panzhifang@wmu.edu.cn

Hongliang Zhang
zhang1hongliang@163.com

Tong Liu
xiongdui12345@163.com

Xiaojia Ye
yxj@lixin.edu.cn

Ali Asghar Heidari
aliasghar68@gmail.com

[1] Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

[2] College of Computer Science and Technology, Jilin University, Changchun 130012, China

[3] Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China

[4] Department of Information Technology, Wenzhou Polytechnic, Wenzhou 325035, China

[5] The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, People's Republic of China

1 Introduction

It is generally known that population-based optimization algorithms can find high-quality solutions based on random operations [1–5, 112–116]. Being a new population optimizer, the salp swarm algorithm (SSA) [6] was created by Mirjalili in 2017. In literature [6], it can be found that SSA is superior to well-known algorithms: harmony search (HS) [7], particle swarm optimization (PSO) [8], gravitational search algorithm (GSA) [9], and flower pollination algorithm (FPA) [10] on CEC-BBOB-2015 test functions, and the multi-objective SSA (MSSA) outperforms the classical multi-objective algorithms: multiobjective particle swarm optimization (MOPSO) [11] and multi-objective evolutionary algorithm based on decomposition (MOEA/D) [12] on IEEE CEC2009 test suits, meanwhile, SSA and MSSA have been used to deal with seven engineering optimization problems. As a swarm algorithm, it has a high potential to approximate the optimal solutions with a satisfying convergence rate and coverage of solution space. In addition, it is much easier to comprehend and implement due to its simple mathematical model and framework. So it has been employed frequently to realize many real-world requests, for instance, feature selection (FS), energy, and environmental protection [13–15]. Lately, Faris et al. have summarized the work on SSA and they adopted it to train extreme learning machine (ELM) significantly [16].

One of the most crucial applications of SSA is FS. Faris et al.[17] built a binary SSA integrating a crossover scheme to deal with FS; 22 well-known UCI datasets tested its performance. Sayed et al. [18] presented a hybrid algorithm by embedding chaos theory into SSA. Mohammad et al. [19] integrated singer chaotic map and a new local search algorithm (LSA) into SSA for FS. Although all the experiment results clearly show that SSA shows promising performance in dealing with FS, the improved SSA variants are aimed at specific datasets. We see they perform badly on some datasets, so the algorithms need to be enhanced further and expand the field of application.

Another critical application of SSA is parameter estimation and result forecast. El-Fergany [20] presented an SSA-based method to evaluate parameters of polymer exchange membrane fuel cells (PEMFCs) to guarantee cognitive modeling, simulation, and control. Zhang et al. [21] applied SSA in parameter estimation for the soil water retention curve. Hussien et al. [22] introduced a K-nearest neighbor (KNN) to SSA for predicting chemical compound activities. Zhao et al. [23] established a novel model named SSA-LSSVM for CO_2 prediction; SSA was used to optimize two parameters in the least-squares support vector machine (LSSVM). Although SSA variants get better results in dealing with parameter identification and result forecast problems, due to the model's performance depends on the given datasets, the precision needs to be further enhanced to be hired in the real application fields.

The SSA was also employed to tackle and recognize solutions to other engineering optimization. Ateya et al. [24] used chaotic maps to enhance SSA and applied it to optimize controllers' numbers and connections between switches and controllers in large scale SDN networks. Yang et al. [25] proposed a memetic SSA (MSSA) for the PV systems in the condition of partial shading. Wang et al. [26] introduced bound estimation into multi-objective SSA to deal with short-term load forecasting problems. Abbassi et al. [27] utilized an SSA-based strategy to obtain the optimal configurations of photovoltaic cell models. Abadi et al. [28] combined SSA and genetic algorithm (GA) for nurses' scheduling and designation in COVID-19 pandemic. Salma et al. [29] adopted both a random mutation operator and an adaptive process to boost the search capability of SSA for solving optimal power flow (OPF) problems. Ewees et al. [30] combined firefly algorithm (FA) with SSA to deal with unrelated parallel machine scheduling problems (UPMSP). Rohit et al. [31] brought the self-adaptive parameters strategy in SSA to boost its convergence performance. Although SSA gets better results in solving real engineering optimization problems, we see the accuracy and efficiency requires to be augmented for satisfying actual needs.

It is observed that the SSA performs better in solving a lot of optimization problems. However, it needs to be improved in terms of convergence precision and rate in dealing with complex optimization problems. A range of scholars applied different local and global search enhanced mechanisms to boost its exploration and exploitation drifts. Ibrahim et al. [32] combined the characteristic of SSA with the feature of PSO, which is named SSAPSO, to improve the global and local search capability of basic SSA. In this framework, if the random value is bigger than 0.5, the SSA's mechanisms are adopted. Otherwise, PSO's mechanisms are applied; they reported that the SSAPSO is better than the simple SSA in dealing with 15 benchmark functions. Furthermore, the SSAPSO is employed to deal with FS problems on the UCI datasets. Rizk-Allah et al. [33] presented five versions of binary SSA with different transformations, and they all have been tested on 18 benchmark functions. The results show that the binary SSA embedding a modified Arctan transformation called BBSA-V has competitive performance. Divya et al. [34] embedded random opposition-based learning, multiple leadership, and simulated annealing into SSA to enhance its performance in solving complex multi-modal problems. Malik et al. [35] employed a novel lifetime scheme to accelerate the convergence performance of SSA. Bhaskar et al. [36] employed Gaussian, Cauchy, and levy-flight mutation schemes to enhance the performance of SSA. Fatima et al. [37] introduced the local best information and differential evolution to improve SSA named MSSA. Nibedan et al. [38] adopted both normal distributed mutation operator and oppositional learning concept to enhance the performance of SSA. Ren et al. [39] used the random replacement strategy and double adaptive weight to improve the performance of SSA. Saafan et al. [40] integrated improved whale optimization algorithm (IWOA) into SSA to boost its performance. Although the above mechanisms have enhanced SSA comparatively, these strategies are usually for the given problems, at the same time the convergence precision and rate of the process requires to be still enhanced.

As mentioned above, though many scholars have done a lot of valuable efforts to improve the property of SSA, we observe its exploitation and exploration propensities need to be further improved. As we experienced, the chaotic map has a good random performance, so it is usually utilized in improving the algorithm's parameters, intensification, and the initialization of the population. Zhang et al. [14] proposed a new SSA variant named CMSSA, three mechanisms: Gaussian mutation, Cauchy mutation, and chaotic exploitative mechanism (CEM) were combing with SSA to boost its performance. The results verify that CMSSA is better than PSO with an aging leader and challengers (ALCPSO) [41], comprehensive learning particle swarm optimizer (CLPSO) [42], self-adaptive differential evolution (SaDE) [43], jDE [44], hybridizing grey wolf optimization (HGWO) [45], and improved grasshopper optimization algorithm (IGOA) [46] on 23 well-known benchmark functions. Emre et al. [47] used a chaotic map to update the control parameter, mutualistic relationship to enhance search performance, and a random technique to enhance the diversity. Liu et al. [48] integrated the chaotic exploitative trends and multi-population structure to enhance the performance of SSA. DE method has a superior searching performance, and it has widely been used to uphold the performance of other algorithms. Rohit et al. [49] adopted the mutation operators to boost the performance of SSA. Zhang et al. [50] combined the improved mutation operation of DE and restart mechanism with SSA to improve its optimization performance.

In this paper, a chaotic SSA with DE (CDESSA) is proposed by adopting chaotic initialization and DE-based mechanisms to help the SSA jump out of the local optimum (LO) and accelerate its convergence rapidity too. As far as we know, these strategies are firstly utilized to advance the effectiveness of SSA regarding exploratory and exploitative drifts. This paper has contributions as follow:

- For one aspect, chaotic initialization is utilized to provide a better initial population; it is conducive to finding better global optima (GO).
- In another aspect, DE is employed to boost the searching capability of each individual to increase the convergence potential and convergence accuracy of SSA.
- Chaotic initialization and DE can help the exploration and exploitation competencies of SSA to make a more suitable balance.
- We provide an efficient solver, called the CDESSA, to deal with functions of IEEE CEC2014, FS problems, constrained engineering optimization to not only solve high-dimensional and multimodal problems but also deal with complex practical application problems.

This work is conducted as below: The basic theory of original SSA was elaborated in Sect. 2. The proposed CDESSA was described in detail in Sect. 3. The experiment results and discussions were explained in Sect. 4. At last, the conclusions and future research plans were given briefly in Sect. 5.

2 Salp swarm algorithm (SSA)

SSA is a bio-inspired optimizer firstly established by Mirjalili in 2017 [6], which is inspired by the swarming activities of salps when they are navigating and foraging for foods in oceans. Interestingly, salps often come together in a chain. This phenomenon can be illustrated in Fig. 1. Though this phenomenon cannot be interpreted by the theory, this interesting phenomenon can be ascribed to gain better movement through both rapid cooperation movement and foraging by Anderson [51].

To establish a mathematical model for the swarm, the salp swarm is divided into two groups: the leader and followers. The salp in the first place of the chain is called leader, so the rest of the salps are treated as followers. Taken literally, the

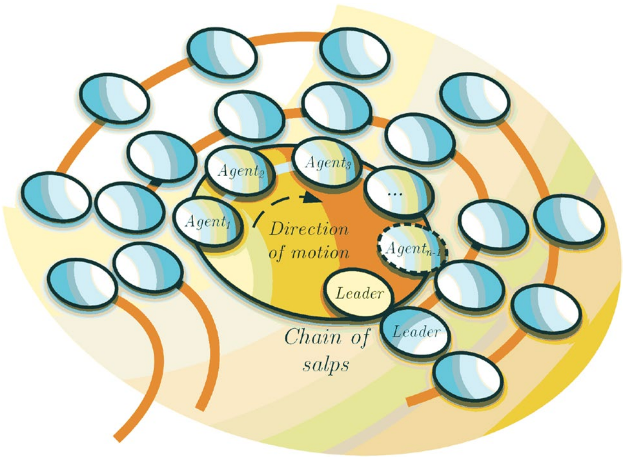


Fig. 1 A demonstration of salp chains

leader leads the direction of the swarm, and the followers follow the leader one after another.

Like other swarm optimizers, the location of salps is defined as a D-dimensional vector where D is the dimension of the given problem. Hence, the position of all salps forms a two-dimension matrix called X. It is also supposed that the food source called F in the search space is the foraging target of the swarm. The following equation updates the position of the leader:

$$X_{1,j} = \begin{cases} F_j + c_1 \times \left(c_2 \times \left(ub_j - lb_j\right) + lb_j\right) & c_3 \geq 0.5 \\ F_j - c_1 \times \left(c_2 \times \left(ub_j - lb_j\right) + lb_j\right) & c_3 < 0.5 \end{cases}, \quad (1)$$

where $X_{1,j}$ shows the position of leader in the jth dimension, F_j is the position of the food source in the jth dimension, ub_j denotes the jth dimension of the upper bound, lb_j indicates the jth dimension of lower bound, c_1, c_2, and c_3 are control parameters.

The rule in Eq. (1) shows that the position of the leader is updated only based on the food source. The parameters c_2 and c_3 are random real numbers that are uniformly generated in the region of [0,1]. The most critical parameter c_1 used to keep the poise between exploration and exploitation drifts in SSA is calculated as follows:

$$c_1 = 2 \times e^{-\left(\frac{4 \times l}{L}\right)^2}, \quad (2)$$

where l means current iteration, meanwhile, L is the maximum iteration.

The location of followers is renewed ed by:

$$X_{i,j} = \frac{1}{2} \times \left(X_{i,j} + X_{i-1,j}\right), i \geq 2, \quad (3)$$

where $X_{i,j}$ means the location of the ith follower salp in the jth dimension.

The flowchart of SSA is demonstrated in Fig. 2. The implemented process of SSA can be depicted as the below steps:

Step 1: Produce the initial salp population randomly within the boundaries of the solution space and evaluate the population and gain the best salp, as the global best solution (*FoodPosition*), and attain its fitness (*FoodFitness*), $l = 1$.

Step 2: If the current iteration l is not more than the maximum iteration L, go to Step 3; otherwise, go to Step 6.

Step 3: Update the parameter c_1 according to Eq. (2).

Step 4: if $i == 1$, replace the location of the leading salp based on Eq. (1); else renew the location of the follower salp based on Eq. (3).

Step 5: Amend the salps according to the upper and lower boundaries of the problem, and evaluate the salp population, if the best salp in the current population is better than the global best solution (*FoodPosition*), then, update the global best solution (*FoodPosition*) by the best salp in the current population; otherwise, the global best solution (*FoodPosition*) keeps unchanged, $l = l + 1$, and go to Step 2.

Step 6: Return the best solution (*FoodPosition*) and its fitness (*FoodFitness*).

3 Proposed chaotic SSA with DE (CDESSA)

The original SSA has some advantages such as quick convergence speed and simple structure. However, when solving more complicated practical optimization problems, it may easily face stagnation problems. The performance of SSA mainly depends on the interactions between the leader agent

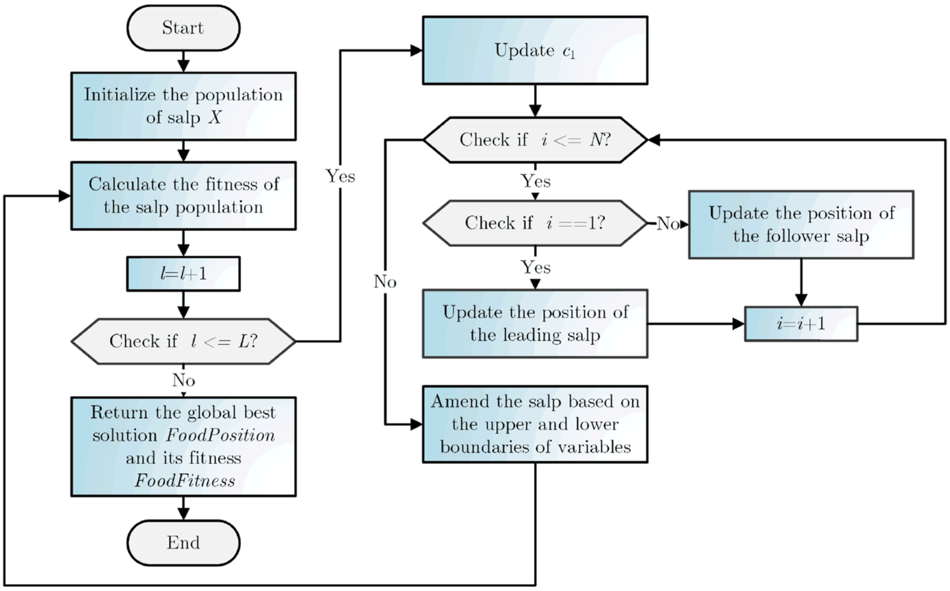

Start
Initialize the population of salp X
Calculate the fitness of the salp population
l=l+1
Check if l <= L?
Yes
No
Return the global best solution FoodPosition and its fitness FoodFitness
End
Update c1
Check if i <= N?
Yes
No
Check if i ==1?
No
Yes
Update the position of the follower salp
Update the position of the leading salp
i=i+1
Amend the salp based on the upper and lower boundaries of variables

Fig. 2 The flowchart of SSA

and follower agents. When a single agent falls into the LO, the agent can jump out of the LO by the full effect of the leading agent. But if most agents sink into the same condition, the whole algorithm will slow down and lock down around the LO finally. This condition provides the opportunity for other strategies to be embedded in SSA, so the performance of SSA will be improved.

3.1 Chaotic initialization mechanism

Literature [52, 53] has pointed out that a well-distributed initial population is always useful. For instance, it will assist the population to converge to a better optimal solution, and it may also augment the convergence speed and quickness of the optimization process. The chaotic map has been widely utilized throughout the initialization period of the optimizers due to the following characteristics: closeness to the initial condition, semi-randomness, and ergodicity. In the initialization phase, for one aspect, the chaotic map is utilized to produce random parameters to replace the uniform or gaussian parameters [54–56]. For another aspect, the chaotic map is adopted to generate a chaotic solution [57–59], as well as the role of opposition-based learning [60].

In this paper, the logistic map is exploited to generate a chaotic solution of the agent in the initialization; the distribution of the logistic map can be denoted by Eq. (4).

$$C(t+1) = \mu \times C(t) \times (1 - C(t));\ t = 1, 2, \ldots, N-1, \quad (4)$$

where μ is coefficient, it is often equal to 4, at this time chaotic map will be produced; t is the index of current chaotic number; N is population size; $C(1)$ indicates a random number in the range $(0, 1)$ that is not equal to 0.25, 0.5, and 0.75; $N-1$ chaotic numbers will be generated based on Eq. (4).

The chaotic solution of the ith agent can be attained by Eq. (5).

$$CX_i = C(t) \times X_i, \quad (5)$$

where $C(t)$ is the tth chaotic numbers; X_i denotes the ith agent; CX_i indicates the chaotic solution of the ith agent.

3.2 Differential evolution mechanism

The DE algorithm is a population-based algorithm and it has been embedded into many optimizers [61, 117]. Its optimization process mainly includes mutation, crossover, and selection, in which mutation is the key factor of DE; it determines the optimization adaptation of DE mainly. Aim to prevent SSA from dropping into premature convergence. The well-known DE is introduced into SSA to generate a new, better candidate solution for each search agent as an intensification tool. The population has N search agents. Each agent is a vector with D dimensions denoted as $X_i = \left(X_{i,1}, X_{i,2}, \ldots, X_{i,D}\right)$.

3.2.1 Mutation

In the mutation process, the mutation operator is carried out by selecting three diverse agents X_{r1}, X_{r2}, and X_{r3} from the current population, where $r1 \neq r2 \neq r3 \neq i$, so a mutant vector U_i can be attained. The mutation operator can be denoted as Eq. (6).

$$U_i = X_{r1} + F \times \left(X_{r2} - X_{r3}\right), \quad (6)$$

where F is the mutation scale factor, it is a vector of D dimensions. Each dimension distributes in the range [0, 1].

In this study, mutation scale factor F can be generated by Eq. (7).

$$F = U\left(F_{\min}, F_{\max}, [1, D]\right), \quad (7)$$

where U stands for uniform distribution, it means generating a vector F with D dimensions. Each dimension is uniformly distributed in the range $\left[F_{\min}, F_{\max}\right]$. In this paper, $F_{\min} = 0.2$ and $F_{\max} = 0.8$.

3.2.2 Crossover

In this process, a trial vector V_i can be generated by carrying out a crossover operator. The crossover operator can be modeled as Eq. (8):

$$V_{ij} = \begin{cases} U_{ij}; & \text{rand}(0,1) < P_{\text{CR}} \quad \text{or} \quad j == j_{\text{rand}} \\ X_{ij}; & \text{Otherwise} \end{cases}, \quad (8)$$

where P_{CR} is a probability factor in determining the diversity of the swarm and prevents the algorithm from dwindling into local minima. In this paper, $P_{CR} = 0.1$. rand is a random number in the range [0, 1]., j_{rand} is an index, which is randomly generated between $\{1, 2, \ldots, D\}$. It ensures that one element of the trial vector V_i comes from mutant vector U_i.

3.2.3 Selection

Selection operator means that the better offspring is chosen between the trial vector V_i and the search agent X_i based on fitness. It can guarantee that the better candidate solution will be retained, so the convergence speed and accuracy of the algorithm will be enhanced. This operator can be denoted as Eq. (9):

$$X_i = \begin{cases} V_i; & f\left(V_i\right) < f\left(X_i\right) \\ X_i; & \text{Otherwise} \end{cases}. \quad (9)$$

3.3 The implementation of CDESSA

The implementation details of CDESSA are introduced in this section. In the original SSA, the search agent moves to the next location based on the leader agent or its previous agent; therefore, a promising solution cannot always be obtained in each time so that the algorithm will trap into early convergence; its convergence speed and accuracy will be prevented to be further improved. To further develop the exploitation and exploration trends of SSA, chaotic initialization mechanism and DE-based mechanism are combined with SSA to form a new enhanced SSA, named CDESSA.

In CDESSA, a chaotic initialization mechanism is utilized to build a better opening population for the subsequent optimization process. Because the quality of the initial population regulates the convergence rapidity and accuracy of the population-based algorithms, a chaotic initialization mechanism guarantees that we keep an initial population as random as possible. This gives the algorithm more chance to escape from the LO and find a more promising solution. A DE-based mechanism is utilized to intensify the exploitation and exploration of the search agent as an intensification technique. This gives the search agents more opportunities to find a more satisfactory solution. Hence, the convergence speed and accuracy of the algorithm will be further enhanced.

The flowchart of CDESSA is shown in Fig. 3. The implemented process of the CDESSA method can be depicted as below:

Step 1: Generate the location of the search agent in the initial population randomly by Eq. (10), FEs = FEs + N.

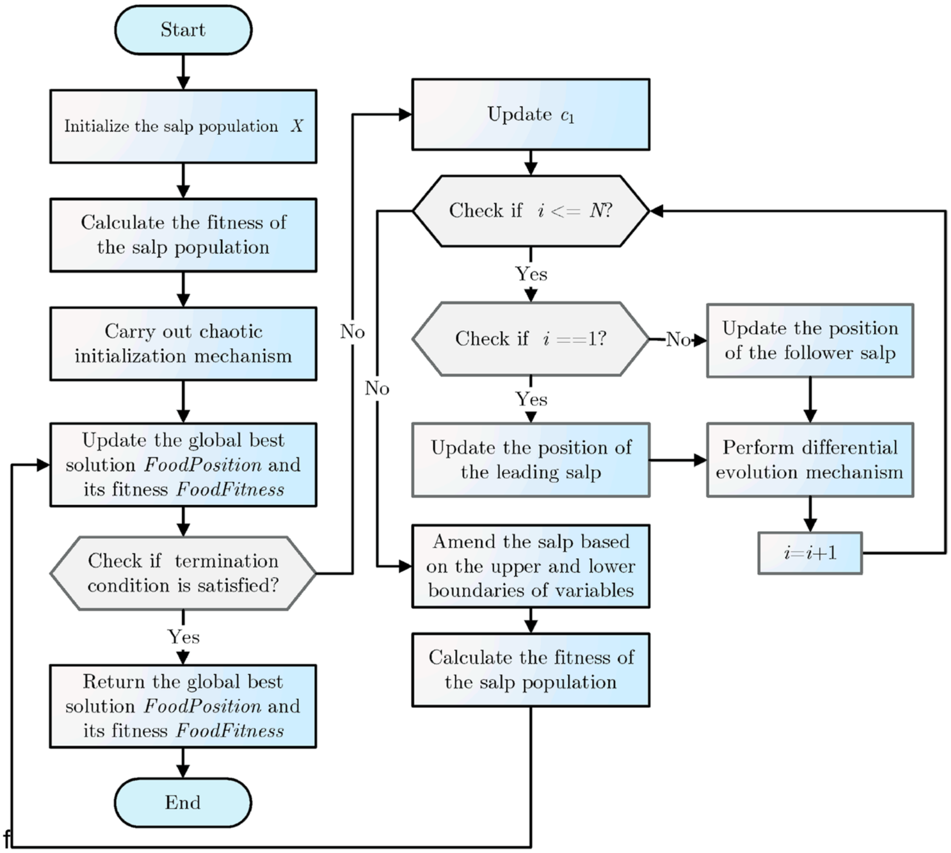
$$X_{i,j} = lb_j + \text{rand}(0,1) \times \left(ub_j - lb_j\right), \tag{10}$$

where $X_{i,j}$ means the jth dimension of the position of the ith agent; lb_j and ub_j denote the jth dimension of the lower bound and upper bound, respectively; rand() produces a random number in the range.

Step 2: Evaluate the fitness of each individual in the initial population; calculate the chaotic solution of the individual in the initial population by Eqs. (4) and (5), thus, the N chaotic solutions can be obtained and form a chaotic population; choose the best N solutions from both the initial population and the chaotic population to update the initial population, repeat this process $N-1$ times and obtain the best initial population to participate in the next iteration process; get the optimal solution (*FoodPosition*), FEs = FEs + $(N-1) \times N$.

Step 3: Initialize the algorithm parameters such as c_1, $F_{\min}$, $F_{\max}$, P_{CR}, in which c_1 can be generated by Eq. (11):

Fig. 3 Flowchart of CDESSA


Start
Initialize the salp population X
Calculate the fitness of the salp population
Carry out chaotic initialization mechanism
Update the global best solution FoodPosition and its fitness FoodFitness
Check if termination condition is satisfied?
Yes
Return the global best solution FoodPosition and its fitness FoodFitness
End
No
Update c1
Check if i <= N?
Yes
No
Check if i ==1?
No
Yes
Update the position of the follower salp
Update the position of the leading salp
Perform differential evolution mechanism
i=i+1
Amend the salp based on the upper and lower boundaries of variables
Calculate the fitness of the salp population

$$c_1 = 2 \times \mathrm{e}^{-\left(\frac{4\times \mathrm{FEs}}{\mathrm{MaxFEs}}\right)^2}, \tag{11}$$

where FEs is the current fitness evolution number; MaxFEs is the maximum fitness evolution number.

Step 4: If $i == 1$, refresh the location of the leader agent by Eq. (1); else renew the location of the follower agents by Eq. (3).

Step 5: Select three different agents from the current population, then carry out a DE mechanism and generate the candidate solution of the search agent by Eqs. (6), (7), (8), update the search agent by Eq. (9), FEs = FEs + N.

Step 6: Amend the position of the agent within the lower and upper bound, evaluate the fitness of each agent in the current population, FEs = FEs + N; if the optimal agent is better than the global optimal solution (*FoodPosition*), update the global optimal solution (*FoodPosition*) by the optimal agent in the current population; else the global optimal solution (*FoodPosition*) keeps unchanged.

Step 7: If the termination condition is met (FEs < MaxFEs), repeat Steps 4–6; otherwise, go to Step 8.

Step 8: Output the global optimal solution (*FoodPosition*) and its fitness (*FoodFitness*).

3.4 The time complexity of CDESSA

The proposed algorithm is made of chaotic initialization mechanism, DE mechanism, and basic SSA, so its computational complexity depends on the chaotic initialization mechanism, DE mechanism, and original SSA. As mentioned earlier, N and MaxFEs signify group size and maximum fitness evolution number, respectively. Similarly, D means the dimension of the given problem, respectively. The computing time complexity of initialization process is $O(N\times D)$, The computing time complexity of chaotic initialization mechanism is $O((N-1)\times N\times D)$. Because the initial population and chaotic initialization consume $N\times N$ fitness evolutions. There are two fitness evolutions in each iteration, so the rest of the iterations are $(\mathrm{MaxFEs}-N\times N)/2$. In each iteration, the computing time complexity of updating the position by basic SSA is $O(N\times D)$, the computing time complexity of the DE mechanism is $O(N\times D)$, the computing time complexity of updating the global optimal solution is $O(N)$. So, the computing time complexity of the proposed algorithm is $O((N\times D+(N-1)\times N\times D)+(N\times D+N\times D+N)\times(\mathrm{MaxFEs}-N\times N)/2)$, which can be simplified to $O((2\times N\times D+N)\times((\mathrm{MaxFEs}-N\times N)/2+N\times N\times D))$.

4 Experimental results

To estimate the performance of CDESSA, a series of experiments is conducted, firstly, the IEEE CEC2014 benchmark suite is used to test the performance of CDESSA in solving high-dimensional and multimodal problems; secondly, FS problems are adopted to evluate the capability of CDESSA in dealing with discrete optimization problems; finally, CDESSA is used to solve four constrained engineering optimization problems. All the experiments can evaluate the performance of CDESSA in handling global optimization problems comprehensively and objectively.

To be fair, as per standards in machine learning works [98–101] all the experiments were performed on the server with Intel (R) Xeon(R) E5-2650 v3 (2.30 GHz) CPU and 20 GB RAM, it runs Windows Server 2008 R2 version operating system, all the codes are implemented by MATLAB R2014b.

4.1 Experiment 1: results for IEEE CEC2014 problems

In this section, CDESSA is compared with the state-of-the-art algorithms on the IEEE CEC2014 benchmark suit. IEEE CEC2014 benchmark suite can be divided into four types such as unimodal, multimodal, hybrid, and composition functions. The summary of the CEC2014 benchmark suite is demonstrated in Table 1. In this experiment, to be fair, the swarm size N was 40, the maximum evaluations (MaxFEs) were set to $D\times 10{,}000$, in which D is problem dimensions, at the same time each method runs 30 times on each function independently.

At the same time, seven competitive improved evolutionary algorithms are chosen to participate in this competition; the parameters of all the involved algorithms are listed in Table 2 in detail. The Wilcoxon singed-rank test [62] was utilized with a 0.05 significance level to measure whether there is a statistical significance between CDESSA and other competitors. In the results, “*w*’’ sign identifies that CDESSA is superior to the competitors, “*t*” symbol denotes a tie and so there is not a statistically significant difference, and “*l*” sign signifies CDESSA is inferior to the peer algorithms, and hence CDESSA exhibits the worse performance.

4.1.1 The impact of chaotic initialization and DE

As it was mentioned in Sect. 3, chaotic initialization and DE mechanism are integrated with the original SSA. To measure the influences of each strategy and acquire the best mechanism combinations for the next experiments. Among these combinations, CSSA indicates only a chaotic initialization mechanism is embedded in the original SSA; DESSA means only the DE mechanism is introduced in basic SSA; CDESSA denotes both chaotic initialization, and DE mechanism are applied to enhance the performance of SSA. All these mechanism combinations are evluated on the CEC2014 benchmark 30D test. Then, the Friedman test

Table 1 IEEE CEC2014 benchmark suit

Index	Function name	Category	Range	Optimum
F1	Rotated high conditioned elliptic function	Unimodal	[− 100, 100]	100
F2	Rotated bent cigar function	Unimodal	[− 100, 100]	200
F3	Rotated discus function	Unimodal	[− 100, 100]	300
F4	Shifted and rotated Rosenbrock’s function	Multimodal	[− 100, 100]	400
F5	Shifted and rotated Ackley’s function	Multimodal	[− 100, 100]	500
F6	Shifted and rotated Weierstrass function	Multimodal	[− 100, 100]	600
F7	Shifted and rotated Griewank’s function	Multimodal	[− 100, 100]	700
F8	Shifted Rastrigin’s function	Multimodal	[− 100, 100]	800
F9	Shifted and rotated Rastrigin’s function	Multimodal	[− 100, 100]	900
F10	Shifted Schwefel’s function	Multimodal	[− 100, 100]	1000
F11	Shifted and rotated Schwefel’s function	Multimodal	[− 100, 100]	1100
F12	Shifted and rotated Katsuura function	Multimodal	[− 100, 100]	1200
F13	Shifted and rotated HappyCat function	Multimodal	[− 100, 100]	1300
F14	Shifted and rotated HGBat function	Multimodal	[− 100, 100]	1400
F15	Shifted and rotated expanded Griewank’s plus Rosenbrock’s function	Multimodal	[− 100, 100]	1500
F16	Shifted and rotated expanded Scaffer’s F6 function	Multimodal	[− 100, 100]	1600
F17	Hybrid function 1 ($N=3$)	Hybrid	[− 100, 100]	1700
F18	Hybrid function 2 ($N=3$)	Hybrid	[− 100, 100]	1800
F19	Hybrid function 3 ($N=4$)	Hybrid	[− 100, 100]	1900
F20	Hybrid function 4 ($N=4$)	Hybrid	[− 100, 100]	2000
F21	Hybrid function 5 ($N=5$)	Hybrid	[− 100, 100]	2100
F22	Hybrid function 6 ($N=5$)	Hybrid	[− 100, 100]	2200
F23	Composition function 1 ($N=5$)	Composition	[− 100, 100]	2300
F24	Composition function 2 ($N=3$)	Composition	[− 100, 100]	2400
F25	Composition function 3 ($N=3$)	Composition	[− 100, 100]	2500
F26	Composition function 4 ($N=5$)	Composition	[− 100, 100]	2600
F27	Composition function 5 ($N=5$)	Composition	[− 100, 100]	2700
F28	Composition function 6 ($N=5$)	Composition	[− 100, 100]	2800
F29	Composition function 7 ($N=3$)	Composition	[− 100, 100]	2900
F30	Composition function 8 ($N=3$)	Composition	[− 100, 100]	3000

Table 2 Parameters settings for the involved algorithms

Algorithm	Parameters
ALCPSO [41]	$w=0.4$; $c_1=c_2=2.0$; $\Theta_0=60$; $T=2$
CLPSO [42]	$w_{\min}=0.2$; $w_{\max}=0.9$; $c=1.496$
DECLS [63]	$F_{\min}=0.2$; $F_{\max}=0.8$; $\mathrm{CR}=0.2$; $m=1500$
WDE [64]	No parameters
ESSA [65]	$r_1=\mathrm{randi}(50)$; $c_1=2\times\exp(-(4\times r_1/\mathrm{MaxFEs})\wedge 2)$; $c_2=\mathrm{rand}(0, 1)$; $c_3=\mathrm{rand}(0, 1)$
CMSSA [14]	$\mu = 4$; $C_1=\mathrm{rand}(0, 1)$ and $C_1\neq 0, 0.25, 0.5, 0.75, 1.00$; $m=1000$ $c_1=2\times\exp(-(4\times\mathrm{FEs}/\mathrm{MaxFEs})\wedge 2)$; $c_2=\mathrm{rand}(0, 1)$; $c_3=\mathrm{rand}(0, 1)$
CSSA [18]	$c_1=2\times\exp(-(4\times\mathrm{FEs}/\mathrm{MaxFEs})\wedge 2)$; $c_2=C\times O_s\times 1\text{-}\ O_s)$; $C=4$; $O_0=0.7$; $c_3=\mathrm{rand}(0, 1)$
CDESSA (ours)	$F_{\min}=0.2$; $F_{\max}=0.8$; $\mathrm{CR}=0.1$; $c_1=2\times\exp(-(4\times\mathrm{FEs}/\mathrm{MaxFEs})\wedge 2)$; $c_2=\mathrm{rand}(0, 1)$; $c_3=\mathrm{rand}(0, 1)$

is employed to measure the differences between all strategies according to the average ranking value. The results of the Friedman test are displayed on Table 3. In Table 3, it can be observed that CDESSA gets the minimum average value of 1.8233 and the best rank No. 1. All these results indicate that the performance of CDESSA outperforms CSSA and DESAA. Based on the results, CDESSA with the best performance comparing with other mechanism combinations

Table 3 Average ranking values of different mechanisms combinations using the Friedman test

Algorithm	Average	Rank	*w*/*l*/*t*
SSA	3.3511	4	25/0/5
CSSA	2.7211	3	16/0/14
DESSA	2.1044	2	7/0/23
CDESSA	1.8233	1	NA

is chosen to take part in the following experiments as the proposed algorithm.

4.1.2 Diversity analysis and balance analysis

To measure the population diversity and the balance between exploration and exploitation of CDESSA and SSA in the search process, the diversity analysis and balance analysis experiment is conducted based on the IEEE CEC2014 benchmark 30*D* test, the population is formed by 40 agents, when iteration reachs 1000 the search process is terminated. The diversity and balance analysis convergence plots are indicated in Figs. 4 and 5, respectively.

In Fig. 4, the horizontal axis indicates the iteration number; the vertical axis denotes the average distance between the individuals in the population. It indicates the individual distribution in the current population. And then, it reveals the population diversity. In Fig. 4, at the initialization phase the mean distance between agents is much bigger because the agents are distrbuted randomly in the solution space, the curve will go down as the iteration increases, it will reach stabilization finally. At the initialization stage, the average distance between the agents of the CDESSA is much bigger than basic SSA owing to the chaotic initialization mechanism performed in the initialization. The chaotic initialization mechanism makes the initial population distributes more uniformly so that the population can search for more potential regions. It indicates that the proposed CDESSA has more diversity at the initial phase. The time at which the average distance between the agents in the CDESSA population reaches the steady-state is later than the original SSA. It can be ascribed to the DE mechanism, which enhances the exploration performance of SSA. It increases the diversity of the population and avoids the population being premature convergence to a certain extent.

In Fig. 5, the horizontal ordinate means the iteration number, the ordinate menas the percentage, the red curve denotes

Fig. 4 Diversity analysis of CDESAA and SSA

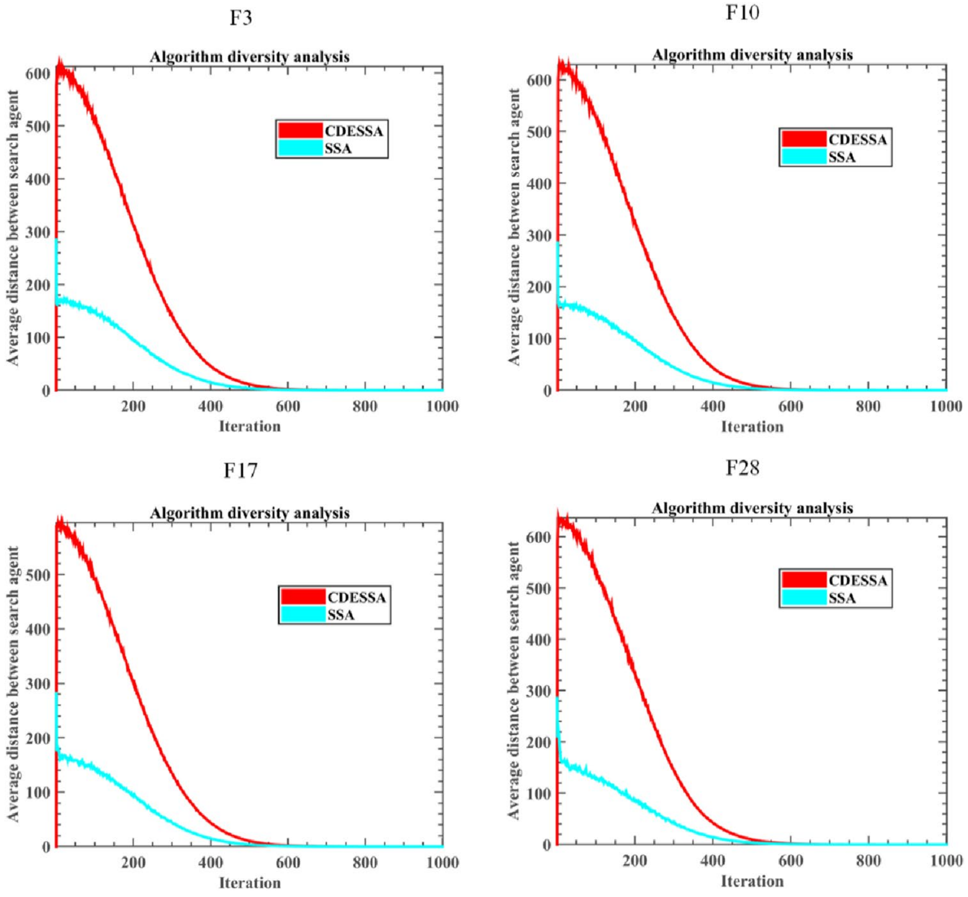
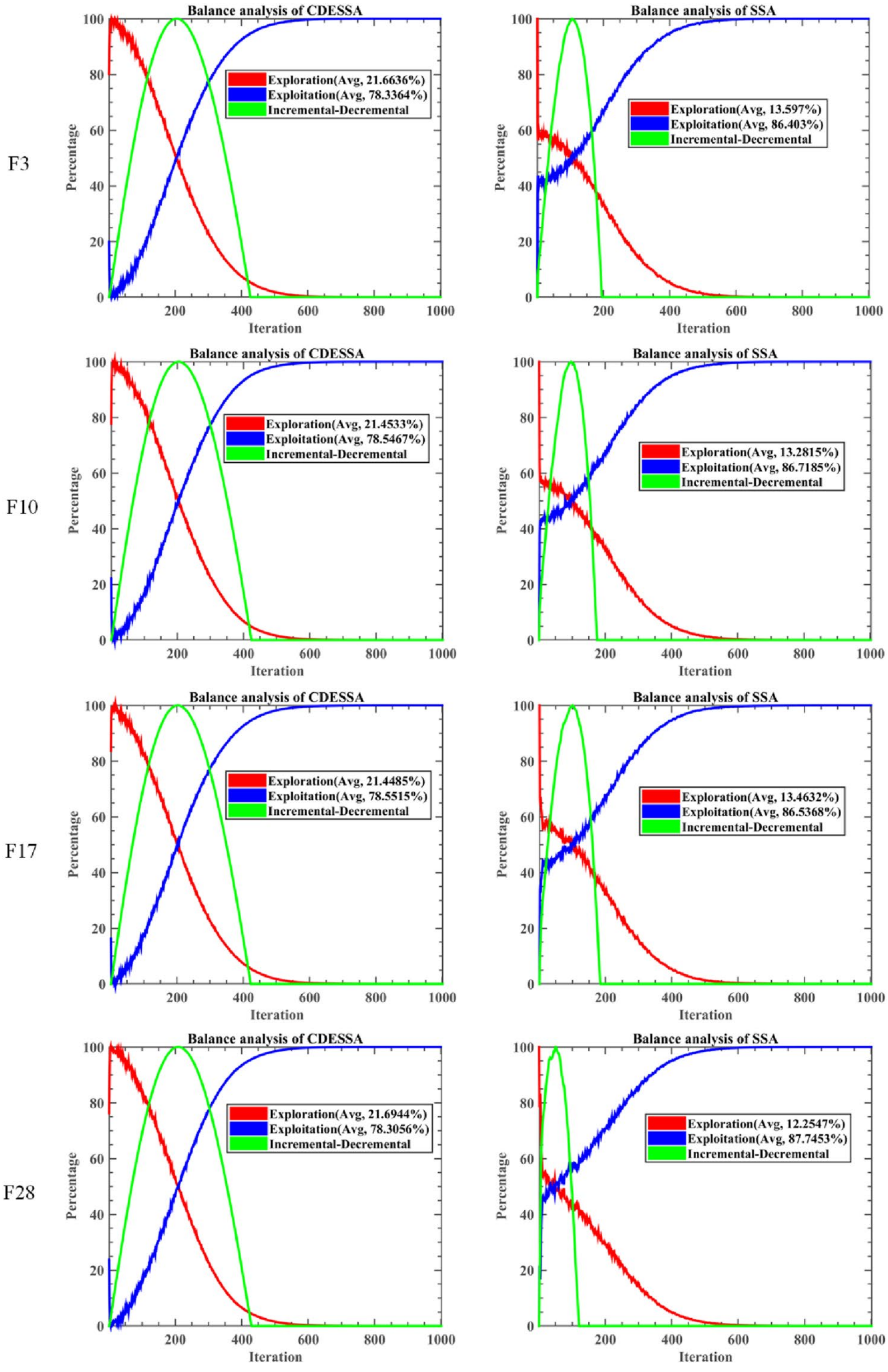


Fig. 5 Balance analysis of CDESAA and SSA



the exploration, the blue curve indicates the exploitation, an cyan incremental-decremental curve is appended in the chart, firstly, it increases from 0 to 100%, when the exploration and exploitation of the algorithm reach balance condition it reaches the biggest value 100%. After this moment, the exploitation curve rises to 100%, but the exploration curve falls to 0, the incremental-decremental curve also drops to 0, this phenomenon matches the optimization cycle of the population-based method, most of the population-based methods mainly performs the global searching stage to accelerate convergence rate in the base phase, in the later stage it carries out the exploitation step to boost the convergence accuracy. From Fig. 5, it can be found that the difference between the exploration and the exploitation in CDESSA is much higher than that in basic SSA, because that the chaotic initialization mechanism enhances the exploration at the initial stage. The moment that the exploration and exploitation of the algorithm archive balance are later in the CDESSA than that in the original SSA, this indicates that the DE mechanism boosts the exploration ability and makes it last longer, so the stable moment will be delayed. In the same way, the average exploration of CDESSA is bigger than the original SSA. This also demonstrates that the chaotic initialization mechanism and the DE mechanism promote the exploration performance of SSA.

4.1.3 Comparison with other state-of-the-art algorithms on 30D test

In this section, CDESSA is in comparison to other state-of-the-art methods at $D=30$. The results of all the competitors for IEEE CEC2014 problems at $D=30$ are shown in Appendix Table 19, which include mean and std values for each function. The overall ranking of all the methods based on the average value is listed in Table 4, the last row of Table 4 shows the total number that CDESSA wins, tie with, and loses to the corresponding competitor, respectively. The results of the Friedman test are exhibited in Table 5. The average execution time of each method on each function is demonstrated in Table 6. The convergence trend pictures of all the involved algorithms are demonstrated in Fig. 6.

Table 4 The overall ranking of all the methods based on average fitness results on IEEE CEC2014 benchmark set at $D=30$

Function	CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	3	2	4	5	1	7	6	8
F2	4	3	2	1	5	7	6	8
F3	5	2	3	4	1	6	7	8
F4	3	5	1	4	2	7	6	8
F5	1	6	3	5	2	7	4	8
F6	1	3	2	5	4	7	6	8
F7	3	4	2	1	5	7	6	8
F8	5	4	1	2	3	7	6	8
F9	1	4	2	5	3	7	6	8
F10	5	4	1	2	3	7	6	8
F11	3	4	1	6	2	7	5	8
F12	1	5	3	4	2	7	6	8
F13	1	5	3	4	2	7	6	8
F14	1	5	3	4	2	7	6	8
F15	1	3	2	4	5	7	6	8
F16	3	4	1	5	2	7	6	8
F17	2	3	4	5	1	7	6	8
F18	3	4	2	5	1	7	6	8
F19	1	5	2	4	3	7	6	8
F20	2	3	4	5	1	6	7	8
F21	4	2	3	5	1	7	6	8
F22	4	5	3	2	1	7	6	8
F23	2	6	5	4	7	3	1	8
F24	2	8	6	4	7	3	1	5
F25	2	8	7	4	6	3	1	5
F26	1	7	4	3	2	6	5	8
F27	2	7	5	4	6	3	1	8
F28	2	7	5	4	6	3	1	8
F29	2	7	4	6	5	3	1	8
F30	2	7	5	6	4	3	1	8
Mean	2.4000	4.7333	3.1000	4.0667	3.1667	5.9667	4.7667	7.8000
Rank	1	5	2	4	3	7	6	8
w/*l*/*t*	NA	21/4/5	18/8/4	23/0/1	17/11/2	30/0/0	23/3/4	30/0/0

Table 5 The average ranking of the peers based on the Friedman test on IEEE CEC2014 benchmark set at $D = 30$

Algorithm	Average ranking	Rank
CDESSA	2.3622	1
ALCPSO	4.5244	5
CLPSO	3.2356	2
DECLS	4.0722	4
WDE	3.3633	3
ESSA	6.0900	7
CMSSA	4.7122	6
CSSA	7.6400	8

Based on the results in Appendix Table 19 and the rangking table Table 4, it can be found that CDESSA outperforms ESSA and CSSA on all 30 functions; CDESSA wins ALCPSO on 21 functions, it loses to ALCPSO on 4 functions (F1, F2, F3, F10), it ties with ALCPSO on other 5 functions; CDESSA is superior to CLPSO on 18 functions, it is inferior to CLPSO on 8 functions (F2, F3, F4, F8, F10, F16, F18, F22); CDESSA outperforms DECLS on 23 functions, it is worse than DECLS on 6 functions (F2, F3, F7, F8, F10, F22), there is no statistically significant difference between CDESSA and DECLS on other one function; Compared with WDE, CDESSA gains more promising results on 17 functions, it loses to WDE on 11 functions (F1, F3, F4, F8, F10, F16, F17, F18, F20, F21, F22), it shows a tie on other 2 functions; Compared to CMSSA, CDESSA shows more competitive performance on 23 functions, it is behind CMSSA on 3 functions (F27, F29, F30), on other 4 functions CDESSA and CMSSA go to deuce. Based on the above analysis, it comes to a conclusion that on most of the Unimodal functions CDESSA

Table 6 The average time of the involved algorithms on IEEE CEC2014 benchmark set at $D = 30$

Function	CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	2.6158E+01	6.0934E+00	1.1700E+01	1.6362E+01	2.1543E+01	5.3274E+00	1.8102E+02	5.1896E+00
F2	2.2375E+01	4.3384E+00	8.1048E+00	1.3437E+01	1.8264E+01	3.9453E+00	1.9535E+02	3.8491E+00
F3	1.5266E+01	3.1044E+00	6.0965E+00	9.6674E+00	1.3985E+01	2.6635E+00	2.1915E+02	2.8954E+00
F4	1.3598E+01	2.8574E+00	5.1829E+00	8.4885E+00	1.2169E+01	2.3400E+00	2.2011E+02	2.6520E+00
F5	1.3838E+01	3.2516E+00	5.7315E+00	8.3320E+00	1.1959E+01	2.5413E+00	2.2083E+02	2.9313E+00
F6	4.2349E+01	3.2175E+01	3.6824E+01	3.9893E+01	4.4447E+01	3.1578E+01	2.8048E+02	3.2221E+01
F7	2.6257E+01	5.5006E+00	9.4708E+00	1.5121E+01	2.1206E+01	5.2697E+00	1.7777E+02	4.9978E+00
F8	2.4877E+01	4.5495E+00	9.4355E+00	1.4783E+01	2.0580E+01	4.2885E+00	1.7900E+02	4.0966E+00
F9	2.6303E+01	5.4866E+00	1.1082E+01	1.5582E+01	2.1088E+01	4.8043E+00	1.8371E+02	4.6613E+00
F10	2.8528E+01	6.4959E+00	1.2195E+01	1.6164E+01	2.1264E+01	5.8870E+00	1.7950E+02	6.0851E+00
F11	2.7532E+01	6.9623E+00	1.3032E+01	1.7792E+01	2.3706E+01	6.9639E+00	1.8715E+02	6.5983E+00
F12	3.1903E+01	1.3420E+01	1.8772E+01	2.1715E+01	2.7090E+01	1.2545E+01	2.1665E+02	1.2839E+01
F13	1.4432E+01	2.9775E+00	5.7122E+00	8.6773E+00	1.2850E+01	2.4669E+00	2.2100E+02	2.7149E+00
F14	2.1869E+01	4.4502E+00	8.8395E+00	1.2944E+01	1.8247E+01	4.1678E+00	1.8874E+02	3.9905E+00
F15	2.7019E+01	5.1387E+00	9.9705E+00	1.5121E+01	2.1627E+01	5.3982E+00	1.7925E+02	5.1044E+00
F16	2.6777E+01	5.8162E+00	1.1726E+01	1.5796E+01	2.1190E+01	5.1272E+00	1.8321E+02	4.9712E+00
F17	2.7632E+01	6.2031E+00	1.1265E+01	1.6324E+01	2.2225E+01	5.9239E+00	1.8220E+02	5.8209E+00
F18	2.0563E+01	4.5141E+00	8.5281E+00	1.2830E+01	1.7317E+01	3.8979E+00	2.1149E+02	3.8558E+00
F19	2.0580E+01	9.4599E+00	1.2648E+01	1.5767E+01	2.0220E+01	8.7995E+00	2.3282E+02	9.2223E+00
F20	2.5844E+01	5.4694E+00	1.1372E+01	1.6251E+01	2.0170E+01	4.8901E+00	1.7055E+02	5.0206E+00
F21	2.6183E+01	6.0882E+00	1.1521E+01	1.6770E+01	2.1628E+01	5.6972E+00	1.7961E+02	5.4658E+00
F22	2.4760E+01	6.1683E+00	1.1559E+01	1.5975E+01	2.0077E+01	5.6296E+00	1.9657E+02	5.3347E+00
F23	3.3324E+01	1.4152E+01	1.8854E+01	2.3892E+01	2.8908E+01	1.4030E+01	2.0033E+02	1.3120E+01
F24	1.9648E+01	7.6456E+00	1.1050E+01	1.4429E+01	1.8700E+01	7.4444E+00	2.2525E+02	7.4948E+00
F25	1.9876E+01	8.8245E+00	1.2054E+01	1.5019E+01	1.8616E+01	7.6232E+00	2.3020E+02	7.9997E+00
F26	5.3570E+01	4.1491E+01	4.5466E+01	4.9744E+01	5.2069E+01	3.8414E+01	3.1190E+02	3.9700E+01
F27	1.7857E+00	1.3830E+00	1.5155E+00	1.6581E+00	1.7356E+00	1.2805E+00	1.0397E+01	1.3233E+00
F28	5.9522E−02	4.6101E−02	5.0518E−02	5.5271E−02	5.7855E−02	4.2682E−02	3.4655E−01	4.4111E−02
F29	1.9841E−03	1.5367E−03	1.6839E−03	1.8424E−03	1.9285E−03	1.4227E−03	1.1552E−02	1.4704E−03
F30	6.6135E−05	5.1224E−05	5.6131E−05	6.1412E−05	6.4283E−05	4.7425E−05	3.8506E−04	4.9012E−05

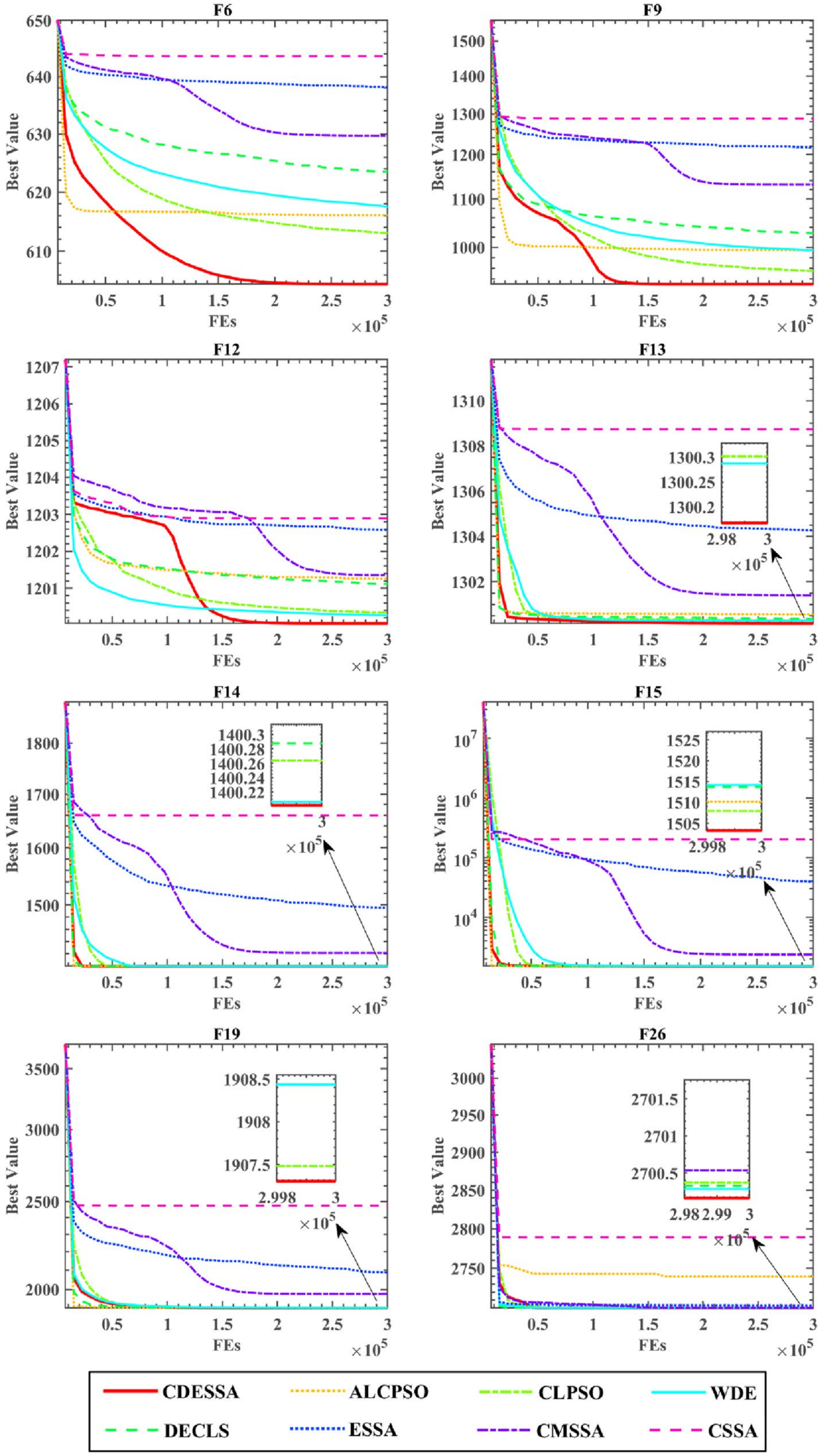


Fig. 6 The convergence graphs of F6, F9, F12, F13, F14, F15, F19, F26 on CEC2014 benchmark set at $D=30$

does not demonstrate more competitive performance than ALCPSO, CLPSO, DECLS, and WDE, at the same time CDESSA does not acquire more promising results on Multimodal functions than WDE, but our proposed algorithm CDESSA shows more competitive performance on Hybrid, and Composition functions, these functions are more difficult to solve than Unimodal and Multimodal functions. In a word, our CDESSA gains the lowest rank value 2.4000 and ranks first, which means CDESSA can beat all the competitors on the IEEE CEC2014 benchmark suit.

The average ranking of all the peers obtained by the Friedman test is shown in Table 5, CDESSA gets the lowest average ranking 2.3622 and No. 1 rank, but the average ranking of No.2 rank CLPSO is 3.2356 who is more than CDESSA, which means CDESSA shows more disadvantage than other competitors.

The average execution time of all the compared algorithms on each function of IEEE CEC2014 problems at $D=30$ is demonstrated in Table 6. From Table 6, it can be found that the time consumption of CMSSA is much bigger than other algorithms due to the chaotic exploitative mechanism (CEM), Gaussian and Cauchy mutation consume more computing time, CSSA and ESSA consume less computing time than other methods owing to it only modifies the parameters and does not introduce extra time consumption, the time consumption of the proposed CDESSA is similar to WDE, it costs less computing time than CMSSA. However, it spends more computing time than CSSA, ESSA, ALCPSO, CLPSO, DECLS. These results indicate that the time complexity of CDESSA is higher than CSSA, ESSA, ALCPSO, CLPSO, DECLS, it is lower than CMSSA; it is equal to WDE.

These results verify that the performance of CDESSA is improved significantly in comparison with the original SSA and other competitors. It also states clearly that the improvements between the CDESSA and other competitors are outstanding. The presented CDESSA performs better due to the following reasons. During the initialization, the chaotic initialization mechanism can obtain a better initial population through the less simulated stochastic process. It ensures that the initial population distributes as uniformly and randomly as possible, literature [52, 53] states clearly that a well-distributed initial population will promote the population to find a better optimal solution faster; during each iteration, the DE mechanism will enhance the exploitation and exploration of each salp as an intensification tool by mutation, crossover, and selection operators, more potential areas will be traversed. All the embedded mechanisms strengthen the exploitation and exploration capacity of the original SSA, so the proposed CDESSA has better convergence accuracy and faster convergence speed, as revealed in Fig. 6 the presented algorithm CDESSA can find a more promising solution quickly.

4.1.4 Scalability test for CDESSA

The scalability test for CDESSA was organized in this section. The dimensionality D is rose to 50 and 100 to estimate the capability of CDESSA in dealing with the benchmark problems with higher dimensions. These state-of-the-art algorithms were utilized as competitors as in Sect. 4.1.3. The comparison results of all the competitors on the IEEE CEC2014 benchmark set at $D=50$ and 100 are exhibited in Appendix Tables 20 and 21, respectively. The overall ranking of all the methods based on the average value on the IEEE CEC2014 benchmark set at $D=50$ and 100 are illustrated in Tables 7 and 8, respectively. From the last row in Tables 7 and 8, it can be seen that the victory function total number of the presented algorithm CDESSA is bigger than the lost function number and the tied function number compared with the competitors. It indicates that the proposed CDESSA outperforms all the peers. From the average Friedman ranking in Table 9, the presented CDESSA achieves the best average ranking value, which is 2.1433 and 1.9211 for $D=50$ and 100, respectively. All the comparison results demonstrate that CDESSA holds distinct advantages in solving higher dimensions to optimize the problem. The CDESSA is far advance than the second one in dealing with higher dimensions problems.

Comparing the average Friedman ranking value for $D=50$ and 100 with $D=30$, it can be found that the average ranking value of the improved CDESSA drops from 2.3622 to 2.1433 and 1.9211 with the increasing dimensionality. It indicates that the enhanced CDESSA holds more advantages in higher dimensionality problems contrary to the competitors; it also means that the introduced mechanisms play a greater role in the high-dimensional problems.

4.2 Experiment 2: results for FS problems

In essence, FS is a kind of discrete optimization problem. This problem aims to choose as few features as possible and get rid of irrespective features to acquire the best classification precision. In this section, the presented CDESSA is adopted to deal with 12 diverse datasets from the UCI machine learning repository [66] comparing with four advanced FS algorithms such as BMFO, BSSA, BWOA, BFOA, SSAPSO [32]. The parameters of all the competitors are set as their original versions in the literature [6, 67–69], respectively. The details of the data sets are described in Table 10, including feature numbers and sample numbers of each dataset.

As we all know, in the beginning, CDESSA is designed for continuous real optimization problems, so it cannot be used to solve FS problems directly. It must be carried out binary transformation as follows.

Table 7 The overall ranking of all the methods based on average fitness results on IEEE CEC2014 benchmark set at $D=50$

Function	CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	1	3	4	5	2	7	6	8
F2	2	4	1	3	5	7	6	8
F3	3	4	2	5	1	6	7	8
F4	3	5	2	1	4	7	6	8
F5	1	6	3	5	2	7	4	8
F6	1	3	2	5	4	7	6	8
F7	2	4	3	1	5	7	6	8
F8	3	4	1	5	2	7	6	8
F9	1	3	2	5	4	7	6	8
F10	5	4	1	3	2	7	6	8
F11	1	4	3	6	2	7	5	8
F12	1	4	3	6	2	7	5	8
F13	1	5	3	4	2	7	6	8
F14	2	5	3	4	1	7	6	8
F15	1	3	2	4	5	7	6	8
F16	3	4	1	5	2	7	6	8
F17	2	3	4	5	1	7	6	8
F18	3	4	1	5	2	7	6	8
F19	3	4	1	5	2	7	6	8
F20	2	3	4	5	1	6	7	8
F21	2	3	4	5	1	7	6	8
F22	4	5	2	3	1	7	6	8
F23	2	7	6	4	8	3	1	5
F24	1	8	6	4	7	3	1	5
F25	1	8	6	4	7	3	1	5
F26	6	7	3	2	1	5	4	8
F27	3	7	6	4	5	2	1	8
F28	2	7	5	4	6	3	1	8
F29	2	7	4	5	6	3	1	8
F30	2	7	5	4	6	3	1	8
Mean	2.2000	4.8333	3.1000	4.2000	3.3000	5.9000	4.7000	7.7000
Rank	1	6	2	4	3	7	5	8
w/*l*/*t*	NA	26/1/3	20/7/3	25/3/2	16/9/5	28/0/2	22/3/5	30/0/0

In the initial stage, every dimension of each agent is assigned 0 or 1 randomly to acquire the binary value. This operation can be denoted as below.

$$X_{i,j} = \begin{cases} 0 & \text{rand}(0,1) < 0.5 \\ 1 & \text{rand}(0,1) \geq 0.5 \end{cases}, \tag{12}$$

where $X_{i,j}$ indicates the *j*th dimension of the value of *i*th agent.

During the optimization process, each dimension of the continuous solution of each agent will be transformed into a binary value by the following transfer functions called the S-shaped transfer function,

$$s = \frac{1}{1 + \mathrm{e}^{-\frac{x}{3}}}, \tag{13}$$

$$x = \begin{cases} \text{posOut} = 1 & \text{rand}(0,1) < s \\ \text{posOut} = 0 & \text{rand}(0,1) \geq s \end{cases}, \tag{14}$$

where *x* indicates each dimension of the position of each agent, posOut means the output binary value of the S-shaped transfer function. These functions realize the aim of transforming continuous real value to binary value.

During the process, each algorithm executes *N* (*N* means the population size of each method) times on each dataset and carries out *K*-fold crossover at each time. During each crossover, the data of each dataset will be divided into three sections: training set, validation set, and test set. First of all, the K nearest neighbor classifier (KNN) is applied to train and classify the data of the training set; then the training model will be verified on the validation

Table 8 The overall ranking of all the methods based on average fitness results on IEEE CEC2014 benchmark set at $D = 100$

Function	CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	1	4	3	6	2	7	5	8
F2	2	4	1	3	5	7	6	8
F3	3	4	1	5	2	7	6	8
F4	3	5	2	1	4	7	6	8
F5	1	5	3	6	2	7	4	8
F6	1	4	2	6	3	7	5	8
F7	3	4	1	2	5	7	6	8
F8	3	4	1	5	2	7	6	8
F9	1	3	2	5	4	7	6	8
F10	4	3	1	5	2	7	6	8
F11	1	4	2	6	3	7	5	8
F12	1	4	3	6	2	7	5	8
F13	1	5	3	4	2	7	6	8
F14	2	5	3	4	1	7	6	8
F15	1	4	2	3	5	7	6	8
F16	3	5	1	6	2	7	4	8
F17	2	3	4	6	1	7	5	8
F18	3	5	1	4	2	7	6	8
F19	1	5	2	4	3	7	6	8
F20	1	4	3	5	2	7	6	8
F21	2	3	4	6	1	7	5	8
F22	1	4	3	5	2	7	6	8
F23	2	8	6	3	7	4	1	5
F24	2	7	6	4	8	3	1	5
F25	2	8	6	4	7	3	1	5
F26	6	8	4	**1**	2	3	5	7
F27	2	7	5	4	6	3	1	8
F28	2	7	5	4	6	3	1	8
F29	2	8	4	5	6	3	1	7
F30	2	7	6	5	4	3	1	8
Mean	2.0333	5.0333	3.0000	4.4333	3.4333	5.9667	4.4667	7.6333
Rank	1	6	2	4	3	7	5	8
w/l/t	NA	27/1/2	20/9/1	27/3/0	21/7/2	29/1/0	22/4/4	30/0/0

Table 9 The average ranking of the involved algorithms achieved by the Friedman test at $D = 50$ and 100

Algorithm	$D = 50$		$D = 100$	
	Average ranking	Rank	Average ranking	Rank
CDESSA	2.1433	1	1.9211	1
ALCPSO	4.6489	5	4.7889	6
CLPSO	3.1722	2	3.1956	2
DECLS	4.2378	4	4.4433	4
WDE	3.4322	3	3.5989	3
ESSA	6.0511	7	6.0100	7
CMSSA	4.6878	6	4.4633	5
CSSA	7.6267	8	7.5789	8

set; finally, the selected features are performed on the test set with the objective of gaining the accuracy value.

In this work, whether each feature will be selected depends on the binary value of the corresponding dimension of the solution. If the value is 0, it indicates this feature is chosen; if the value is 1, it means this feature is abandoned. Each FS scheme that corresponds to the position of the agent is estimated based on the fitness acquired by the method. The fitness *fit* can be calculated by Eq. (15) as follows:

$$\text{fit} = \alpha \times (1 - \text{accuracy}) + \beta \times \frac{N_i}{N}, \tag{15}$$

where accuracy means the accuracy acquired by the algorithm on the validation set. N indicates the total feature

Table 10 The details of the selected datasets from UCI

Dataset	Feature numbers	Sample numbers
Breastcancer	10	699
BreastEW	31	569
Exactly	14	1000
primary-tumor	18	399
heart	14	270
M-of-n	14	1000
SpectEW	23	267
CongressEW	17	435
CTG3	22	2126
Cleveland_heart	14	303
Brain_Tumor2	10,368	50
Tumors_14	15,010	308

number in the corresponding dataset. N_i is the feature number which is selected by the algorithm. The parameters α and β are weight coefficients of the accuracy and feature number, respectively.

In this experiment, the maximum iteration is 50. The population size is 10. Each algorithm runs ten times randomly, Wilcoxon singed-rank test [62] was also utilized with a 0.05 significance level to measure whether there is a statistical significance between CDESSA and other competitors.

The comparison results of fitness obtained by competitors are shown in Appendix Table 22. From Appendix Table 22, it can be found that CDESSA acquires the best results on Breastcancer, primary-tumor, heart, M-of-n, SpectEW; at the same time, CDESSA obtains second best results on BreastEW, Exactly, Cleveland_heart, Tumors_14. The boxplot of fitness obtained by the competitors on datasets is demonstrated in Fig. 7 also supports this conclusion, and in Fig. 7, it can be seen that the fitness fluctuation range of CDESSA is smaller than the competitors on most datasets. The ranking values of peers illustrated in Table 11 indicate that our proposed algorithm CDESSA gets the best ranking value 2.5667 and first rank No.1. The results of the fitness state clearly that the presented CDESSA acquires the best FS set on the datasets. The convergence pictures of the compared algorithms on fitness are shown in Fig. 8. The convergence graphs indicate our proposed CDESSA holds a faster convergence speed against the competitors.

The comparison results of errors acquired by peers are demonstrated in Appendix Table 23. It can be found that CDESSA obtains the lowest error value on Exactly, primary-tumor, heart, SpectEW, CongressEW, Tumors_14; meanwhile, CDESSA obtains second best results on Breastcancer, M-of-n. The boxplot of error value obtained by the peers on datasets is illustrated in Fig. 9 also affirms this conclusion; at the same time, it can be found that the error volatility of CDESSA is lower than the compared algorithms on most of the datasets. The ranking values of all the compared algorithms shown in Table 12 make clear that our presented algorithm CDESSA obtains the best ranking value 2.5708 and first rank No.1. The results of the error value state clearly that the FS scheme found by CDESSA holds the highest accuracy rate on the classifier at the datasets.

The comparison results of the feature number gained by compared algorithms are illustrated in Appendix Table 24. From Appendix Table 24, it can be concluded that CDESSA obtains the smallest feature number on Exactly, primary-tumor, heart, CongressEW, Cleveland_heart; at the same time, CDESSA obtains second best results on BreastEW, M-of-n, CTG3. The boxplot of feature number acquired by the peers on datasets is demonstrated in Fig. 10 also indicates this, meanwhile, it can be discovered that CDESSA has a lower fluctuation on the feature number than rivals on most datasets in Fig. 10. The ranking values of competitors shown in Table 13 indicate that the proposed algorithm CDESSA obtains the best ranking value 2.7042 and No.1 rank value. The results of feature numbers indicate that CDESSA acquires the shortest FS set on the datasets. All the feature number results indicate that the FS set gained by CDESSA contains the fewest uncorrelated feature information.

4.3 Experiment 3: results for constrained engineering optimization problems

In this section, our proposed CDESSA is adopted to deal with five constrained engineering optimization problems. When solving these problems, for CDESSA the population size is set to 50. The maximum iteration is 2000, CDESSA runs 100 times randomly. The statistical results of all the compared algorithms are extracted from the corresponding references due to the best results reported by researchers.

4.3.1 Tension/compression spring design problem

Finding solutions using a developed approach needs a logical modeling process [70–73]. The spring design problem is a widespread case in the engineering design field. The objective of the spring design problem is to devise a minimum weight $(f(x))$ compression spring holding three variables: wire diameter (d), mean coil diameter (D), and the number of active coils (P) [74–76].

This problem holds the mathematical model as below:

consider $x = \left[x_1, x_2, x_3\right] = [d, D, P]$.

object $\min f\left(\vec{x}\right) = x_1^2 x_2\left(x_3 + 2\right)$.

subject to:

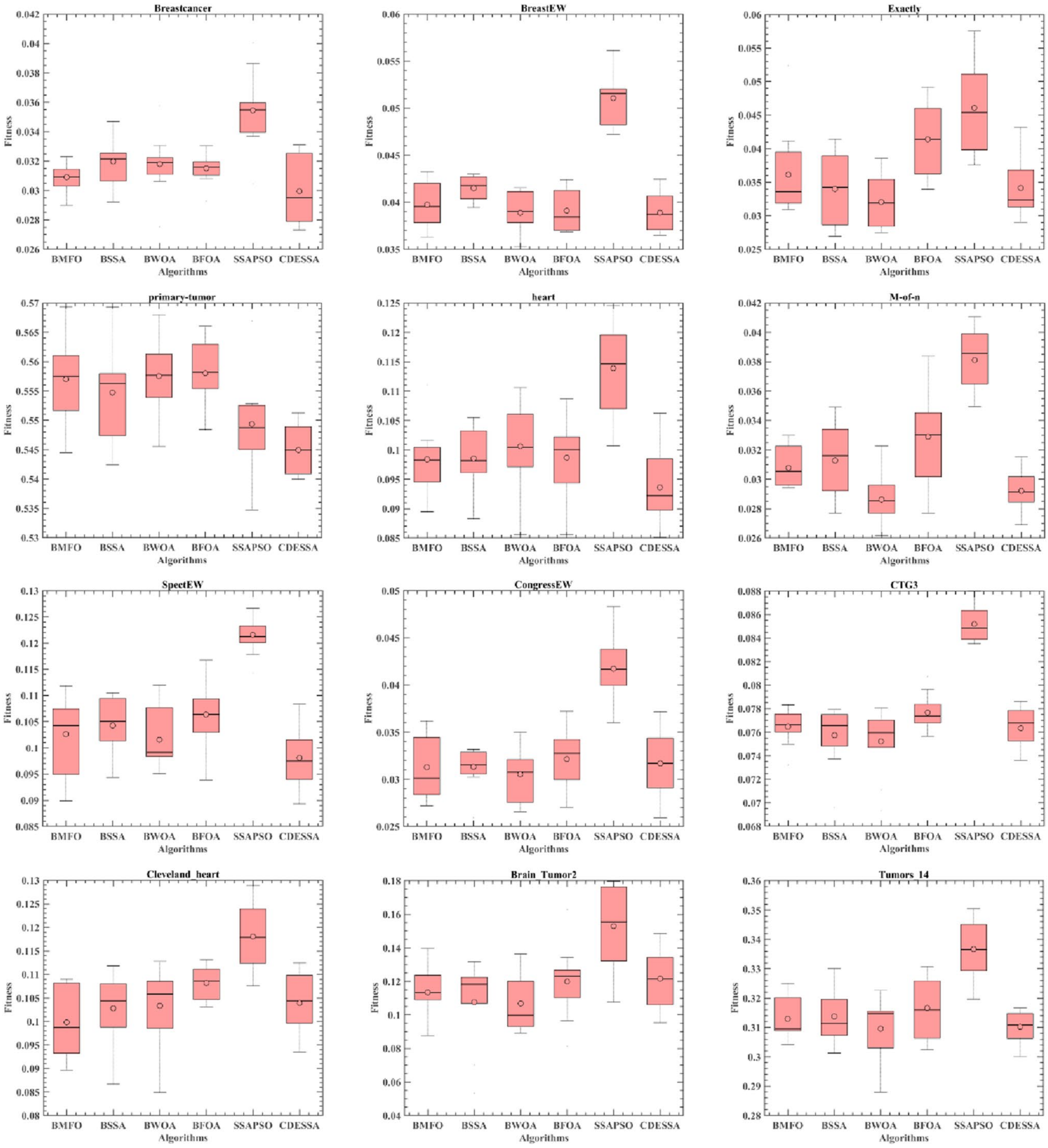


Fig. 7 The boxplot of the fitness among competitors on the datasets

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5180 x_1^2} \le 0$$

Table 11 The ranking value of the compared algorithms achieved by the Friedman test on fitness value

Dataset	BMFO	BSSA	BWOA	BFOA	SSAPSO	CDESSA
Breastcancer	2.3	3.6	3.5	3.4	5.6	2.6
BreastEW	3	4.4	2.4	2.7	6	2.5
Exactly	3.2	2.5	2.1	4.85	5.45	2.9
Primary-tumor	4.1	3.8	4.2	4.7	2.4	1.8
Heart	3.3	3.1	3.8	3	5.6	2.2
M-of-n	3.5	3.3	1.65	4.4	5.9	2.25
SpectEW	3.1	3.3	2.85	3.65	6	2.1
CongressEW	2.7	3.1	2.85	3.5	6	2.85
CTG3	2.9	2.8	2.6	3.8	6	2.9
Cleveland_heart	2.6	2.7	2.9	4.1	5.6	3.1
Brain_Tumor2	2.9	3	2.9	3.8	5.2	3.2
Tumors_14	3	3.1	2.9	3.7	5.9	2.4
Mean	3.05	3.225	2.8875	3.8	5.4708	2.5667
Rank	3	4	2	5	6	1

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^3 x_3} \le 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

variable scope $0.05 \le x_1 \le 2.00, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15.0$.

The best solution obtained by CDESSA for the tension/compression spring design problem is shown in Table 14. From Table 14, it can be found that CDESSA is superior to both original SSA and other competitors in solving tension/compression spring design problems. All the results indicate that CDESSA maintains a more competitive performance for tension/compression spring design problems, in which GA3 is the worst performer, ESSA performs the second best.

4.3.2 Welded beam design problem

Design a welded beam with minimum fabrication cost is another kind of typical engineering design problem. This problem holds four design variables, which are reported in detail in related refs [81].

To solve this problem, the corresponding mathematical model is built as below:

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

object $\min f\left(\vec{x}\right) = 1.10471x_1^2 + 0.04811x_3x_4\left(x_4 + 14.0\right)$

subject to:

$$g_1\left(\vec{x}\right) = \tau(x) - \tau_{\max} \le 0$$

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

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

$$g_4\left(\vec{x}\right) = 0.10471x_1^2 + 0.04811x_3x_4\left(x_4 + 14.0\right) - 5 \le 0$$

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

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

$$g_7\left(\vec{x}\right) = P - P_{c(x)} \le 0$$

variable scope $0.1 \le x_1, x_4 \le 2.0, 0.1 \le x_2, x_3 \le 10.0$

where $\tau\left(\vec{x}\right) = \sqrt{\tau'^2 + 2\tau'\tau''\frac{x_2}{2R} + \tau''^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\tau(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_4x_3^3}, P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000 \text{ lb}, L = 14 \text{ in}, E = 30 \times 10^6 \text{ psi}, G = 12 \times \text{ psi}$$

$$\tau_{\max} = 13,600 \text{ psi}, \sigma_{\max} = 30,000 \text{ psi}, \delta_{\max} = 0.25 \text{ in.}$$

The best solution gained by CDESSA is shown in Table 15. From Table 17, it can be concluded that CDESSA further outperforms all the compared

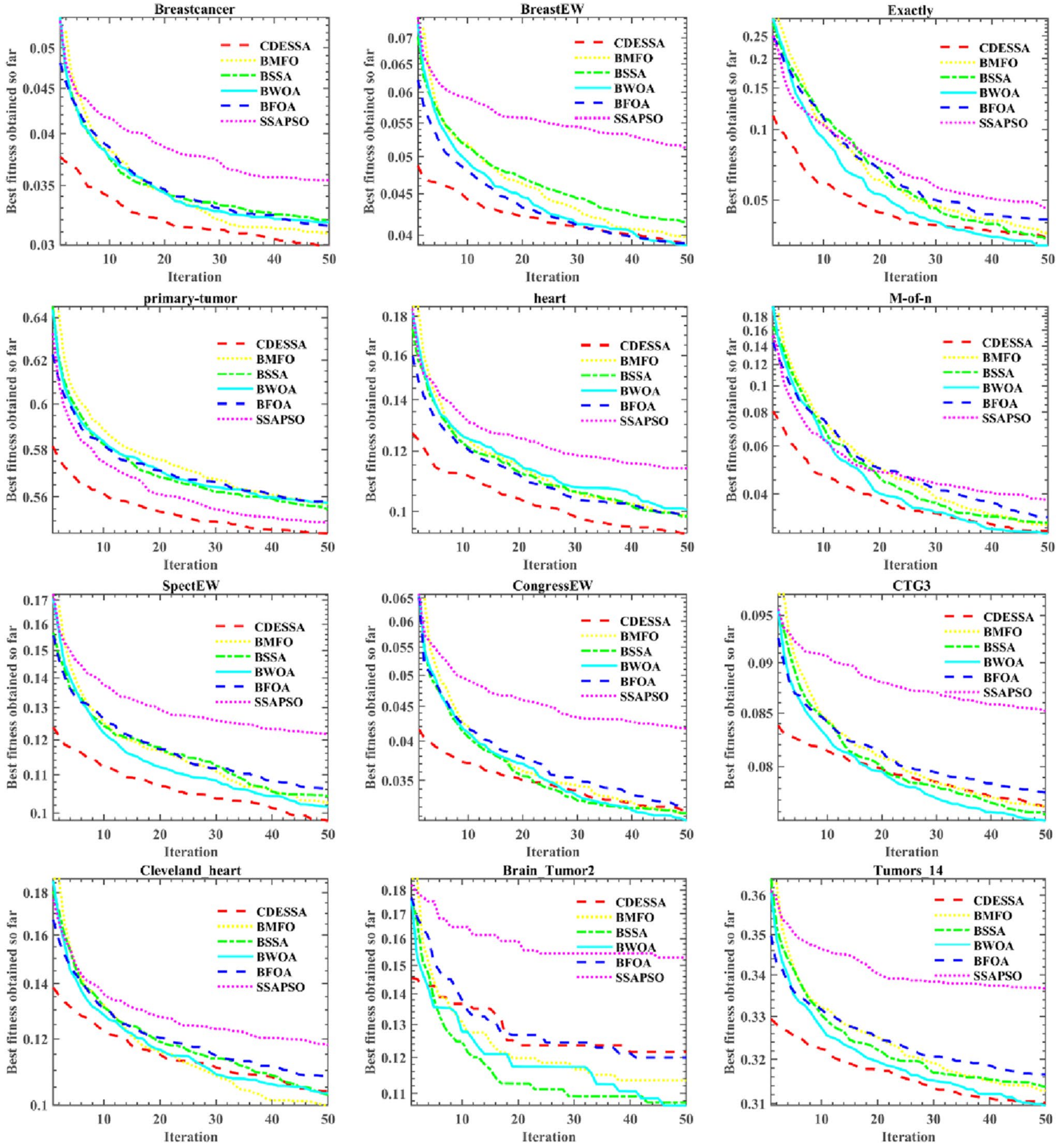


Fig. 8 The convergence graphs of the competitors on the datasets

methods. The results got by other methods are all bigger than 1.72, but the proposed CDESSA is less than 1.7, the difference between CDESSA and other peers is obvious. It means the proposed CDESSA can find a more promising solution within the variable boundaries and restraints.

4.3.3 Pressure vessel design problem

This design problem aims to devise a pressure vessel using a minimum total cost under the constraints of material cost, forming cost, and welding cost. This problem contains four variables: the shell thickness (T_s), the head

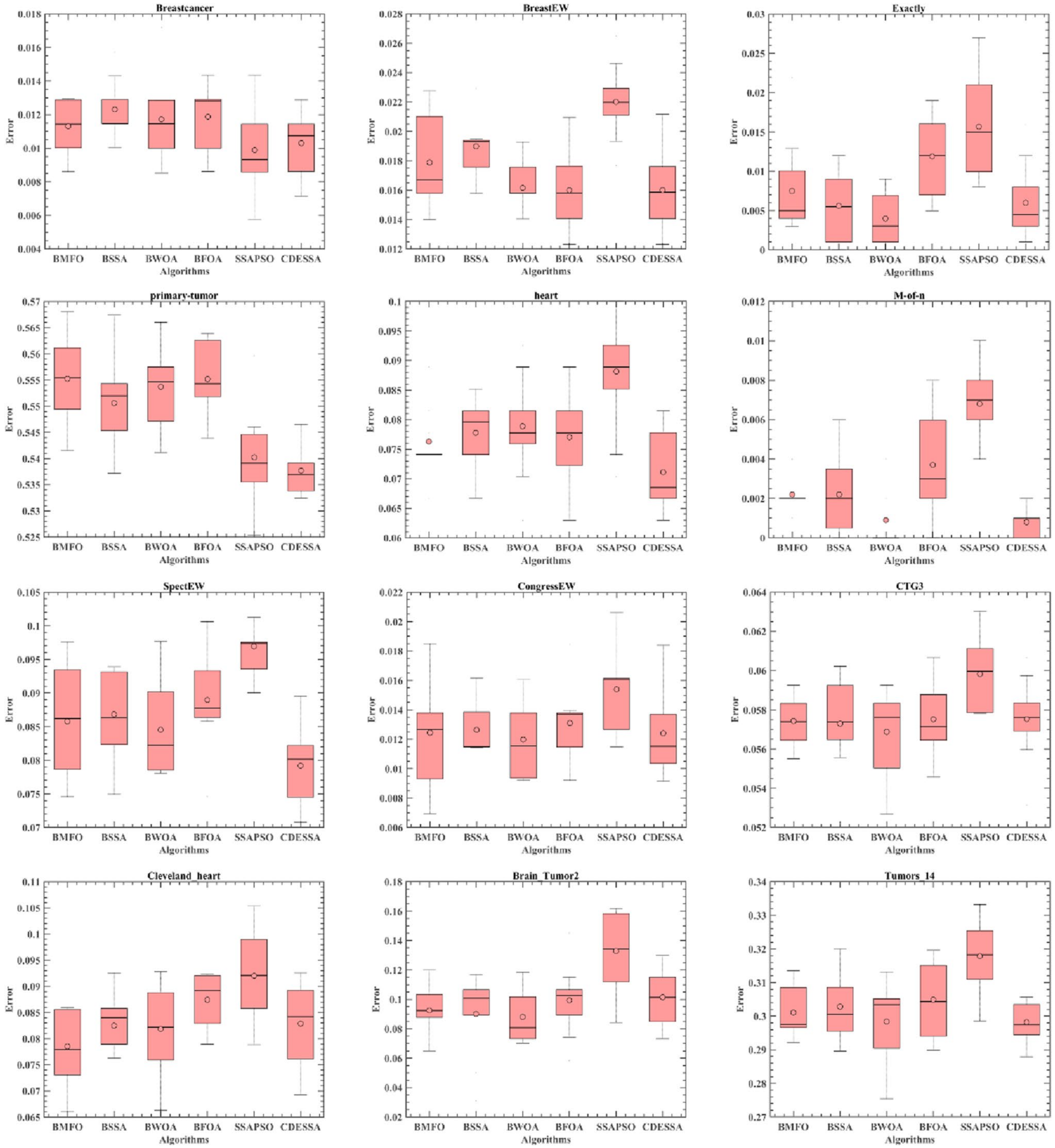


Fig. 9 The boxplot of the error among competitors on the datasets

thickness (T_h), inner radius (R), the cylindrical section length (L); in which, both T_s and T_h are the integral multiples of 0.625 in [84].

The corresponding mathematical model of this problem is built as follows:

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

object $\min f(\vec{x}) = 0.6224x_1x_2x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$.

subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$$

Table 12 The ranking value obtained by the pees based on the Friedman test on error value

Dataset	BMFO	BSSA	BWOA	BFOA	SSAPSO	CDESSA
Breastcancer	3.6	4.4	3.8	3.95	2.65	2.6
BreastEW	3.45	4.2	2.55	2.85	5.6	2.35
Exactly	3.25	2.55	2.05	4.75	5.35	3.05
Primary-tumor	4.6	3.8	4	4.8	2.4	1.4
Heart	3.25	3.35	3.8	3.3	5.1	2.2
M-of-n	3.7	3.2	2.1	4.25	5.7	2.05
SpectEW	3.3	3.2	2.95	3.85	5.7	2
CongressEW	3.45	3.45	3.2	3.35	4.8	2.75
CTG3	3	3.5	2.9	2.9	5.3	3.4
Cleveland_heart	2.4	3.1	3.4	3.7	4.9	3.5
Brain_Tumor2	2.9	3.3	3	3.6	5.05	3.15
Tumors_14	3.1	3.5	3	4.1	4.9	2.4
Mean	3.3333	3.4625	3.0625	3.7833	4.7875	2.5708
Rank	3	4	2	5	6	1

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

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

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

variable scope $0 \leq x_1, x_2 \leq 100, 10 \leq x_3, x_4 \leq 200$.

The best solution gained by the proposed CDESSA is in Table 16. As shown in Table 16, it can be observed that the best result gained by CDESSA is much smaller than other peers, quality of the solution acquired by the competitors is more than 6000, our method CDESSA is only 5453.2428, the difference between CDESSA and other compared approaches is more than 600, so it means CDESSA is much better than other methods. All these results indicate that our method has a much more competitive performance in solving this problem.

4.3.4 Three-bar design problem

The goal of this problem is to design a truss using three bars with the minimum weight, it is a classical constraining optimization problem, it includes two variables: the cross-sectional areas of the bars (A_1, A_2) [89, 90].

To solve this problem its corresponding mathematical model can be constructed as follows:

consider $x = [x_1, x_2] = [A_1, A_2]$.

object $\min f(\vec{x}) = \left(2\sqrt{2x_1} + x_2\right) \times l$.

subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0$$

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

variable scope $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$. where $l = 100$ cm, $P = 2$ KN/cm^2, $\sigma = 2$ KN/cm^2.

The optimal results got by CDESSA and other competitors are demonstrated in Table 17. Based on the values in Table 17 it can be observed that DEDS, PSO-DE, SSA, and CDESSA obtain the same best result 263.8958434, Ray and Sain acquire the worst result 264.3. CDESSA shows better competitiveness in comparison to other evolutionary algorithms. So it can be used to solve the three-bar design problem effectively.

4.3.5 Multiple disk clutch brake design problem

This design problem is a kind of classical discrete optimization problem, its goal is to design a multiple disk clutch brake with both the minimum mass of the multiple disk clutch brake systems. This problem involves five variables: internal diameter (r_i), external diameter (r_o), the thickness of the disc (t), activating force (F), the number of frictional-force(Z). At the same time, the value range of r_i and r_o is [54, 73] and [83], respectively; their step size is both one. The variable t changes in the region [1], and its step size is 0.5. The value range of F is [600, 1000] and its step size is 10. The variable Z changes in the region, and its step size is one. It means the type of all the variables value is discrete [83, 95].

The corresponding mathematical model of this problem can be built as below:

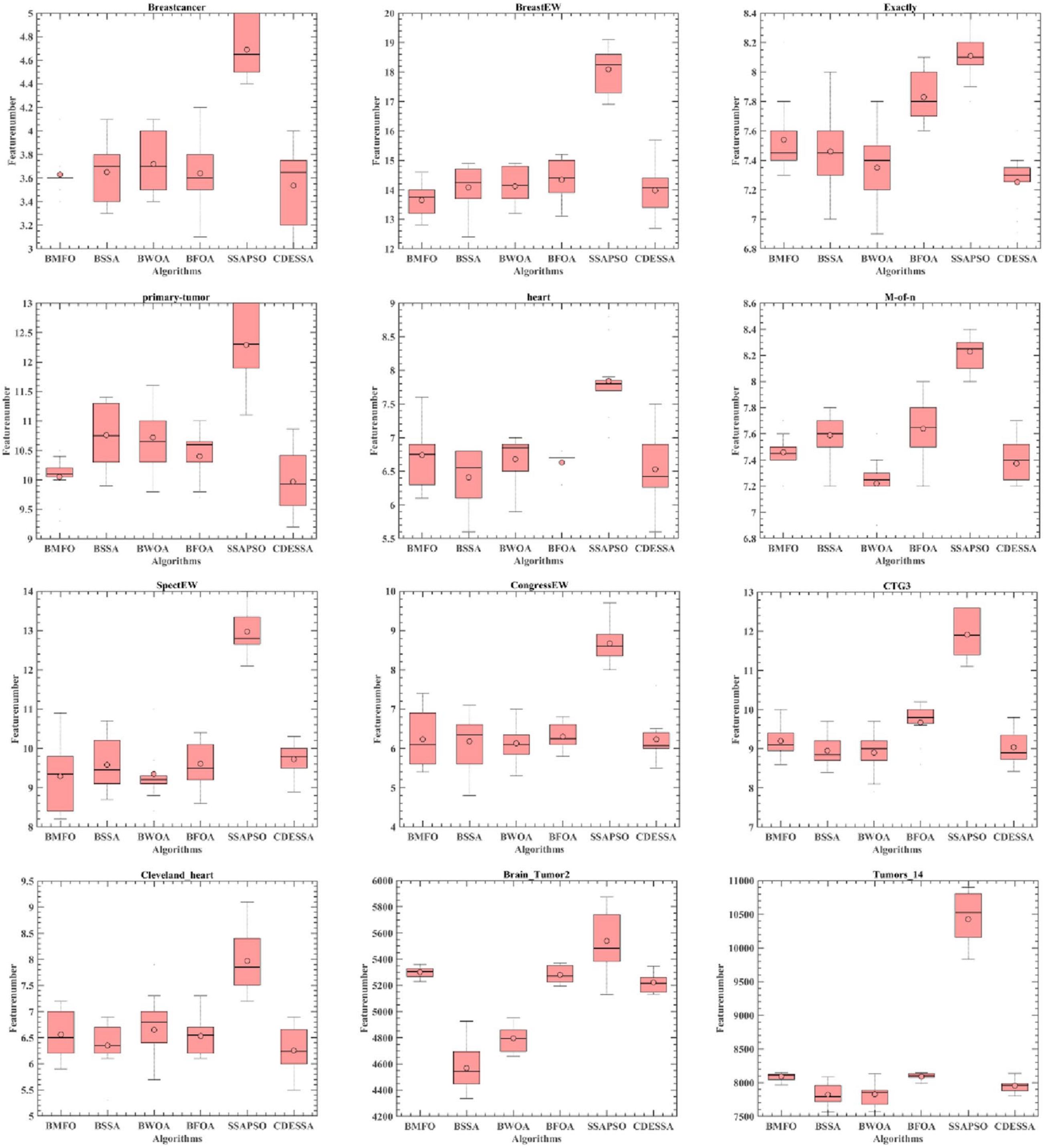


Fig. 10 The boxplot of the feature number among competitors on the datasets

consider $\vec{x} = [x_1, x_2, x_3, x_4, x_5] = [r_i, r_o, t, F, Z]$.

object $\min f(\vec{x}) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)\rho$.

subject to:

$$g_1(\vec{x}) = x_2 - x_1 - \Delta R > 0$$

$$g_2(\vec{x}) = L_{\max} - (x_5 + 1)(x_3 + \delta) \geq 0$$

$$g_3(\vec{x}) = p_{\max} - p_{\mathrm{rz}} \geq 0$$

Table 13 The ranking value obtained by the pees based on the Friedman test on feature number

Dataset	BMFO	BSSA	BWOA	BFOA	SSAPSO	CDESSA
Breastcancer	3.05	3.05	3.25	2.9	6	2.75
BreastEW	2.15	3.2	3	3.7	6	2.95
Exactly	3.15	2.8	2.6	4.85	5.65	1.95
Primary-tumor	1.95	3.8	3.85	3.2	6	2.2
Heart	3.15	2.4	3.5	3.1	5.9	2.95
M-of-n	3	3.85	1.7	4	6	2.45
SpectEW	3	2.95	2.6	3.1	6	3.35
CongressEW	2.9	3.15	2.75	3.35	6	2.85
CTG3	3.4	2.55	2.2	4.15	6	2.7
Cleveland_heart	3.15	2.85	3.7	3.1	6	2.2
Brain_Tumor2	4.7	1	2	4.1	5.7	3.5
Tumors_14	4.4	1.9	1.9	4.2	6	2.6
Mean	3.1667	2.7917	2.7542	3.6458	5.9375	2.7042
Rank	4	2	3	5	6	1

$$g_4(\vec{x}) = p_{\max} V_{\mathrm{sr,max}} - p_{\mathrm{rz}} V_{\mathrm{sr}} \geq 0$$

$$g_5(\vec{x}) = V_{\mathrm{sr,max}} - V_{\mathrm{sr}} \geq 0$$

$$g_6(\vec{x}) = M_{\mathrm{h}} - sM_{\mathrm{s}} \geq 0$$

$$g_7(\vec{x}) = T \geq 0$$

$$g_8(\vec{x}) = T_{\max} - T \geq 0$$

where $M_{\mathrm{h}} = \frac{2}{3}\mu x_4 x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2}\mathrm{N\ mm}$, $\omega = \frac{\pi n}{30}$ rad/s, $A = \pi\left(x_2^2 - x_1^2\right)$ mm^2, $p_{rz} = \frac{x_a}{A}$ N/mm^2.

$V_{\mathrm{sh}} = \frac{\pi R_{\mathrm{sr}} n}{30}$ mm/s, $R_{\mathrm{sr}} = \frac{2}{3}\frac{x_2^3 - x_1^3}{x_2^2 - x_1^2}$ mm, $\Delta R = 20$ mm, $L_{\max} = 30$ mm, $\mu = 0.5$

$p_{\max} = 1\mathrm{MPa}$, $= 0.0000078$ kg/mm^3, $V_{\mathrm{srmax}} = 10$ m/s, $s = 1.5$, $T_{\max} = 15\mathrm{s}$.

$n = 250rpm$, $M_{\mathrm{s}} = 40$ Nm, $M_{\mathrm{f}} = 3\mathrm{Nm}$, $I_z = 55$ kg m^2, $\delta = 0.5$ mm, $r_{i,\min} = 60$ mm.

$r_{i,\max} = 80\mathrm{mm}$, $r_{0,\min} = 90\mathrm{mm}$, $r_{0,\max} = 110\mathrm{mm}$, $t_{\min} = 1.5\mathrm{mm}$, $t_{\max} = 3\mathrm{mm}$.

$F_{\max} = 1000\mathrm{N}$, $Z_{\max} = 9$.

This problem is a discrete combinatorial optimization problem, it needs to satisfy more constraints, so it is harder to solve. The best solutions acquired by CDESSA and other compared methods are shown in Table 18. It can be observed that CDESSA gets the best solution with the lowest function value 0.235242; CBA ranks second; WCA and TLBO acquire the same result 0.313656, PVS obtains the worst results 0.313660. It can be found that CDESSA is superior to other competitors and CDESSA holds more advantages.

Table 14 The optimal schemes acquired by the peers for the tension/compression spring design problem

Algorithm	Optimum variables			Optimum weight
	d	D	P	
GA3 [75]	0.051989	0.363965	10.890522	0.0126810
CPSO [77]	0.051728	0.357644	11.244546	0.0126740
PSO [52]	0.015728	0.357644	11.244543	0.0126747
NM-PSO [78]	0.051620	0.355498	11.333272	0.0126300
DE [79]	0.051609	0.354714	11.410831	0.0126702
SSA [6]	0.051207	0.345215	12.004032	0.0126763
ESSA [80]	0.051719	0.357434	11.247123	0.0126653
CDESSA	0.051691	0.356776	11.285558	0.0126652

5 Conclusions and future directions

In this study, the performance of SSA is enhanced by chaotic initialization and DE mechanism. The presented CDESSA was firstly estimated on IEEE CEC2014 benchmark problems to test the efficacy in solving high-dimensional and multimodal problems. Then, it is utilized to deal with feature selection problems and constrained engineering optimization problems. During the experiments, the proposed CDESSA is compared with the state-of-the-art algorithms. The results and evaluations verified the improved performance of CDESSA in terms of quality of results and convergence trends, and the Friedman test also shows that CDESSA is significantly superior to those competitors. The results verify that the chaotic initialization and DE mechanism enhance the exploration and exploitation capability of the original SSA effectively. The proposed mechanism in CDESSA can improve the equilibrium between the diversification and

Table 15 The optimal results acquired by the peers for the welded beam design problem

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
GA3 [75]	0.205986	3.471328	9.020224	0.206480	1.728226
CPSO [77]	0.202369	3.544214	9.048210	0.205723	1.728024
CAEP [82]	0.205700	3.470500	9.036600	0.205700	1.724852
NM-PSO [78]	0.205830	3.468338	9.036624	0.205730	1.724717
WCA [83]	0.205728	3.470522	9.036620	0.205729	1.724856
SSA [6]	0.205700	3.471400	9.036600	0.205700	1.724910
ESSA [80]	0.197198	3.485213	8.980946	0.208288	1.723317
CDESSA	0.205719	3.253264	9.036825	0.205729	1.695277

Table 16 The best results acquired by the peers for the pressure vessel design problem

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
Branch-bound [85]	1.125000	0.625000	58.291000	43.690000	7198.0428
Lagrangian multiplier [84]	1.125000	0.625000	58.291000	43.690000	7198.0428
CDE [86]	0.812500	0.437500	42.098400	176.637600	6059.7340
GA3 [75]	0.812500	0.437500	42.097400	176.654000	6059.9463
CPSO [77]	0.812500	0.437500	42.091300	176.746500	6061.0777
HPSO [87]	0.812500	0.437500	42.098400	176.636600	6059.7143
G-QPSO [88]	0.812500	0.437500	42.098400	176.637200	6059.7208
CDESSA	0.750000	0.375000	41.966408	178.306673	5453.2428

Table 17 The optimal schemes acquired by the peers for the three-bar design problem

Algorithm	Optimum variables		Optimum weight
	A_1	A_2	
Ray and Sain [91]	0.795000	0.395000	264.3
DEDS [92]	0.78867513	0.40824828	263.8958434
PSO-DE [93]	0.7886751	0.4082482	263.8958434
MBA [90]	0.7885650	0.4085597	263.8958522
CS [94]	0.78867	0.40902	263.9716
SSA [6]	0.788665414258065	0.408275784444547	263.8958434
CDESSA	0.788675	0.408248	263.8958434

Table 18 The optimal results acquired by the peers for the multiple disk clutch brake problem

Algorithm	Optimum variables					Optimum cost
	r_i	r_o	t	F	Z	
CBA [96]	80.000000	90.000000	3.000000	1000.000000	2.000000	0.263684
WCA[83]	70.000000	90.000000	1.000000	910.000000	3.000000	0.313656
PVS [95]	70.000000	90.000000	1.000000	980.000000	3.000000	0.313660
TLBO [97]	70.000000	90.000000	1.000000	810.000000	3.000000	0.313656
CDESSA	70.000000	90.000000	1.000000	970.000000	2.000000	0.235242

intensification cores of SSA and mitigate its convergence and stagnation shortcomings.

In future works, our research plans to focus on the following themes: firstly, introduce the surrogate model to CDESSA to reduce the computation cost; secondly, CDESSA will be applied to handle other challenging optimization problems, such as multi-objective problems, and dynamic optimization problems; then, chaotic initialization and DE strategy will be adopted to enhance the performance of other representative computational intelligence algorithms for engineering optimization problems, such as monarch butterfly optimization (MBO) [102], earthworm optimization algorithm (EWA) [103], elephant herding optimization (EHO) [104], moth search (MS) algorithm [105]; finally, the advanced adaptive DE [106, 107] and adaptive distributed DE [108–110] a lgorithms will be arranged to further enhance the performance of the proposed method.

Appendix A

See Tables 19, 20, 21, 22, 23, and 24.

Table 19 The comparison of the statistical results obtained by all the algorithms for IEEE CEC2014 benchmark set at $D = 30$

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	Mean	5.1178E+06	3.8183E+06	8.6948E+06	3.5424E+07	1.2191E+06	4.4948E+08	1.7684E+08	1.4524E+09
	Std	2.4857E+06	4.4221E+06	2.8766E+06	7.2489E+06	3.8228E+05	9.1378E+07	7.9380E+07	3.1774E+08
F2	Mean	1.1616E+04	2.7288E+03	3.3033E+02	2.8621E+02	1.9791E+05	2.8886E+10	7.0948E+09	8.0997E+10
	Std	4.9199E+03	4.0431E+03	2.8546E+02	3.7650E+02	5.5043E+04	5.5721E+09	3.6092E+09	8.6653E+09
F3	Mean	2.4218E+03	5.0318E+02	5.1807E+02	8.2948E+02	3.0678E+02	5.8719E+04	6.5821E+04	8.6033E+04
	Std	1.7644E+03	4.7619E+02	4.1198E+02	5.4118E+02	2.0947E+00	6.5709E+03	9.2802E+03	4.2412E+03
F4	Mean	5.0427E+02	5.2013E+02	4.6732E+02	5.1888E+02	4.7405E+02	4.5246E+03	1.1231E+03	1.7201E+04
	Std	3.0615E+01	4.3643E+01	2.1343E+01	1.4877E+01	2.1154E+01	1.2952E+03	3.1329E+02	3.0831E+03
F5	Mean	5.2000E+02	5.2084E+02	5.2035E+02	5.2066E+02	5.2010E+02	5.2094E+02	5.2063E+02	5.2100E+02
	Std	5.4914E−04	4.1263E−02	3.3276E−02	6.4879E−02	1.4489E−02	6.6935E−02	1.0610E−01	7.5178E−02
F6	Mean	6.0451E+02	6.1605E+02	6.1304E+02	6.2352E+02	6.1752E+02	6.3816E+02	6.2971E+02	6.4361E+02
	Std	2.0009E+00	3.2295E+00	1.3128E+00	1.1726E+00	1.4102E+00	1.0527E+00	3.7748E+00	1.9596E+00
F7	Mean	7.0000E+02	7.0002E+02	7.0000E+02	7.0000E+02	7.0022E+02	9.8088E+02	7.4964E+02	1.4290E+03
	Std	5.3021E−03	2.5217E−02	1.0773E−04	4.7851E−05	4.2756E−02	4.4217E+01	1.9734E+01	9.0238E+01
F8	Mean	8.2965E+02	8.2590E+02	8.0000E+02	8.0057E+02	8.2039E+02	1.0915E+03	9.9710E+02	1.1486E+03
	Std	1.2340E+01	1.1249E+01	2.1111E−14	6.8025E−01	2.4843E+00	1.4627E+01	2.3029E+01	2.6091E+01
F9	Mean	9.3058E+02	9.9528E+02	9.5484E+02	1.0284E+03	9.9459E+02	1.2162E+03	1.1313E+03	1.2881E+03
	Std	8.7011E+00	2.8439E+01	9.1513E+00	9.8285E+00	1.0061E+01	2.0212E+01	2.6263E+01	2.5424E+01
F10	Mean	2.5326E+03	1.6146E+03	1.0002E+03	1.0592E+03	1.1743E+03	7.8095E+03	5.5847E+03	8.8347E+03
	Std	5.3345E+02	2.8425E+02	2.0865E−01	3.0006E+01	7.2388E+01	2.5732E+02	6.6503E+02	4.5561E+02
F11	Mean	3.3447E+03	4.1266E+03	3.2931E+03	6.2535E+03	3.3239E+03	8.4011E+03	6.1996E+03	9.4236E+03
	Std	5.9149E+02	6.8264E+02	2.3655E+02	2.6771E+02	3.7297E+02	3.4310E+02	7.0523E+02	4.6617E+02
F12	Mean	1.2000E+03	1.2013E+03	1.2003E+03	1.2011E+03	1.2003E+03	1.2026E+03	1.2014E+03	1.2029E+03
	Std	2.6773E−02	7.0762E−01	5.0381E−02	1.6030E−01	3.5121E−02	2.9375E−01	4.3493E−01	5.8209E−01
F13	Mean	1.3002E+03	1.3006E+03	1.3003E+03	1.3004E+03	1.3003E+03	1.3043E+03	1.3014E+03	1.3088E+03
	Std	4.3694E−02	8.8522E−02	4.0031E−02	5.1422E−02	3.4639E−02	3.0038E−01	8.3066E−01	6.3670E−01
F14	Mean	1.4002E+03	1.4005E+03	1.4003E+03	1.4003E+03	1.4002E+03	1.4956E+03	1.4213E+03	1.6598E+03
	Std	2.9654E−02	2.9438E−01	2.9997E−02	4.3346E−02	1.6199E−02	1.2457E+01	1.2326E+01	3.4827E+01
F15	Mean	1.5033E+03	1.5102E+03	1.5080E+03	1.5137E+03	1.5142E+03	3.9173E+04	2.3612E+03	2.0237E+05
	Std	7.1186E−01	4.2158E+00	1.2449E+00	1.0237E+00	1.8570E+00	1.7312E+04	1.0336E+03	5.7712E+04
F16	Mean	1.6117E+03	1.6117E+03	1.6101E+03	1.6118E+03	1.6107E+03	1.6129E+03	1.6121E+03	1.6130E+03
	Std	5.1713E−01	4.6002E−01	3.8732E−01	3.3162E−01	4.2076E−01	2.1211E−01	5.7342E−01	2.7261E−01
F17	Mean	3.5523E+05	6.1469E+05	9.5440E+05	1.9938E+06	6.4131E+03	1.2874E+07	1.0017E+07	2.0851E+08
	Std	2.3001E+05	9.8434E+05	3.5717E+05	8.2585E+05	1.1857E+03	5.1141E+06	5.5887E+06	8.7289E+07
F18	Mean	3.0982E+03	9.2892E+03	1.8912E+03	2.8229E+04	1.8849E+03	5.2829E+08	4.1053E+05	6.9709E+09
	Std	1.1338E+03	6.8859E+03	4.3581E+01	2.0136E+04	1.2122E+01	1.9103E+08	1.5381E+06	2.3652E+09

Table 19 (continued)

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F19	Mean	1.9073E+03	1.9099E+03	1.9075E+03	1.9091E+03	1.9084E+03	2.0875E+03	1.9772E+03	2.4720E+03
	Std	1.2202E+00	2.3585E+00	7.3849E−01	6.5911E−01	8.8385E−01	4.6144E+01	4.3889E+01	1.1743E+02
F20	Mean	2.5654E+03	3.1253E+03	4.5585E+03	5.8161E+03	2.0615E+03	3.5192E+04	3.6802E+04	8.3994E+05
	Std	3.6691E+02	6.4931E+02	1.3313E+03	2.0172E+03	1.2106E+01	1.4011E+04	1.7025E+04	9.4277E+05
F21	Mean	1.3905E+05	9.8407E+04	1.0113E+05	3.6529E+05	3.3914E+03	4.0021E+06	1.3701E+06	1.0476E+08
	Std	1.1740E+05	1.0701E+05	5.9635E+04	1.4589E+05	1.9269E+02	1.4150E+06	9.9496E+05	6.7098E+07
F22	Mean	2.5501E+03	2.6415E+03	2.4015E+03	2.3905E+03	2.3561E+03	3.2611E+03	2.9349E+03	4.1500E+04
	Std	1.5187E+02	1.6510E+02	8.7300E+01	8.6361E+01	8.2231E+01	2.0342E+02	2.5402E+02	9.0925E+04
F23	Mean	2.5000E+03	2.6152E+03	2.6152E+03	2.5000E+03	2.6153E+03	2.5000E+03	2.5000E+03	2.6315E+03
	Std	4.8880E−10	3.4930E−03	1.9961E−05	1.4838E−03	6.5050E−03	6.6696E−04	0.0000E+00	2.9750E+02
F24	Mean	2.6000E+03	2.6346E+03	2.6242E+03	2.6000E+03	2.6305E+03	2.6000E+03	2.6000E+03	2.6008E+03
	Std	8.2131E−07	6.9787E+00	2.8352E+00	1.2568E−02	2.3898E+00	1.9544E−03	0.0000E+00	3.0240E−01
F25	Mean	2.7000E+03	2.7105E+03	2.7080E+03	2.7000E+03	2.7070E+03	2.7000E+03	2.7000E+03	2.7000E+03
	Std	1.0097E−11	2.9481E+00	9.0397E−01	1.8428E−05	8.6860E−01	7.8508E−06	0.0000E+00	1.3814E−02
F26	Mean	2.7002E+03	2.7396E+03	2.7004E+03	2.7003E+03	2.7003E+03	2.7033E+03	2.7005E+03	2.7891E+03
	Std	4.8157E−02	6.3586E+01	6.5513E−02	5.4189E−02	3.5895E−02	3.1199E−01	3.0667E−01	2.5344E+01
F27	Mean	2.9000E+03	3.4269E+03	3.1133E+03	2.9829E+03	3.1159E+03	2.9000E+03	2.9000E+03	4.6425E+03
	Std	4.0005E−06	2.3071E+02	4.7968E+00	1.7252E+02	3.9843E+00	9.5188E−05	0.0000E+00	3.5225E+02
F28	Mean	3.0000E+03	4.3573E+03	3.7004E+03	3.0649E+03	3.7869E+03	3.0000E+03	3.0000E+03	1.0600E+04
	Std	8.0742E−06	5.4016E+02	3.2056E+01	1.9809E+02	3.0236E+01	1.7517E−04	0.0000E+00	1.6905E+03
F29	Mean	3.1000E+03	1.5025E+06	3.8571E+03	7.1376E+03	4.1467E+03	3.8087E+03	3.1000E+03	7.1757E+07
	Std	1.0010E−02	3.9131E+06	1.0846E+02	9.9666E+03	1.3439E+02	4.7289E+02	0.0000E+00	1.6145E+08
F30	Mean	3.2000E+03	1.3056E+04	6.5089E+03	7.3542E+03	5.8091E+03	3.2530E+03	3.2000E+03	9.9014E+06
	Std	4.2556E−03	1.4091E+04	8.4703E+02	1.2939E+03	7.1908E+02	5.6418E+01	0.0000E+00	5.9535E+06

Table 20 The comparison of the statistical results obtained by all the algorithms for IEEE CEC2014 benchmark set at $D=50$

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	Mean	4.9717E+06	1.3860E+07	1.6105E+07	1.7817E+08	5.1869E+06	1.2768E+09	2.8028E+08	8.0607E+09
	Std	1.1071E+06	1.2575E+07	4.8790E+06	2.5782E+07	1.2849E+06	2.6815E+08	1.1331E+08	2.0410E+09
F2	Mean	2.4341E+03	1.1335E+04	2.3598E+02	8.8604E+03	4.5509E+06	7.9764E+10	1.7121E+10	1.7177E+11
	Std	1.7794E+03	9.8814E+03	3.1860E+01	7.5909E+03	1.0651E+06	7.3787E+09	4.1677E+09	1.4287E+10
F3	Mean	2.6810E+03	3.9672E+03	1.9200E+03	4.7623E+03	7.1866E+02	1.1283E+05	1.4386E+05	2.0410E+05
	Std	1.3086E+03	3.2273E+03	9.2032E+02	1.6114E+03	7.7945E+01	9.6557E+03	2.2301E+04	5.5604E+04
F4	Mean	5.1857E+02	5.5072E+02	5.0210E+02	4.9769E+02	5.2376E+02	1.6895E+04	2.5573E+03	4.8780E+04
	Std	3.2124E+01	5.1645E+01	6.9195E+00	1.2985E+00	1.3693E+01	4.2758E+03	9.6743E+02	6.2749E+03
F5	Mean	5.2000E+02	5.2099E+02	5.2044E+02	5.2094E+02	5.2018E+02	5.2112E+02	5.2069E+02	5.2116E+02
	Std	8.7611E−05	6.8421E−02	2.9670E−02	6.6778E−02	2.0161E−02	4.6617E−02	1.2686E−01	4.3715E−02
F6	Mean	6.0654E+02	6.3646E+02	6.3019E+02	6.5423E+02	6.3760E+02	6.6740E+02	6.5698E+02	6.7686E+02
	Std	2.1157E+00	3.7348E+00	1.7134E+00	1.6261E+00	1.7642E+00	1.9326E+00	4.6951E+00	2.7664E+00
F7	Mean	7.0000E+02	7.0002E+02	7.0000E+02	7.0000E+02	7.0072E+02	1.5189E+03	8.7646E+02	2.2393E+03
	Std	2.2570E−03	2.9020E−02	1.3482E−03	8.7157E−05	7.3273E−02	8.9320E+01	4.7746E+01	1.2078E+02
F8	Mean	8.5432E+02	8.6053E+02	8.0000E+02	8.8905E+02	8.4584E+02	1.3779E+03	1.1792E+03	1.4521E+03
	Std	1.0967E+01	1.6686E+01	3.6566E−14	8.6441E+00	3.5131E+00	1.8239E+01	3.2398E+01	2.9142E+01
F9	Mean	9.7625E+02	1.1167E+03	1.0293E+03	1.2139E+03	1.1217E+03	1.5389E+03	1.3524E+03	1.6603E+03
	Std	1.8384E+01	5.2566E+01	1.4174E+01	1.5608E+01	2.3528E+01	2.4766E+01	5.7128E+01	4.4152E+01
F10	Mean	4.5061E+03	2.8314E+03	1.0005E+03	2.5130E+03	1.7142E+03	1.4090E+04	9.7217E+03	1.5638E+04
	Std	7.3326E+02	4.9080E+02	3.6802E−01	3.0465E+02	1.1210E+02	4.2865E+02	7.3561E+02	6.6454E+02

Table 20 (continued)

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F11	Mean	5.4622E+03	7.0212E+03	6.0840E+03	1.2356E+04	6.0156E+03	1.4763E+04	1.0780E+04	1.6078E+04
	Std	8.2249E+02	8.2361E+02	3.4059E+02	4.2241E+02	2.9035E+02	3.8872E+02	1.2156E+03	5.6881E+02
F12	Mean	1.2000E+03	1.2015E+03	1.2003E+03	1.2019E+03	1.2003E+03	1.2035E+03	1.2017E+03	1.2039E+03
	Std	1.6879E−02	7.4723E−01	3.8324E−02	1.3953E−01	3.1793E−02	2.7726E−01	5.4827E−01	6.3396E−01
F13	Mean	1.3003E+03	1.3006E+03	1.3004E+03	1.3004E+03	1.3004E+03	1.3059E+03	1.3021E+03	1.3090E+03
	Std	7.0850E−02	1.0539E−01	4.6607E−02	5.7507E−02	4.0805E−02	3.3150E−01	1.0097E+00	4.5530E−01
F14	Mean	1.4003E+03	1.4007E+03	1.4003E+03	1.4003E+03	1.4002E+03	1.6117E+03	1.4390E+03	1.7458E+03
	Std	3.4372E−02	3.3983E−01	2.4591E−02	1.4305E−01	1.9778E−02	1.9335E+01	1.2122E+01	2.2414E+01
F15	Mean	1.5064E+03	1.5261E+03	1.5184E+03	1.5306E+03	1.5380E+03	8.5442E+05	2.2421E+04	8.1612E+06
	Std	1.1733E+00	8.1862E+00	2.4146E+00	1.9423E+00	4.0054E+00	3.0121E+05	2.0084E+04	3.5799E+06
F16	Mean	1.6207E+03	1.6212E+03	1.6186E+03	1.6217E+03	1.6196E+03	1.6225E+03	1.6220E+03	1.6229E+03
	Std	7.6452E−01	6.4079E−01	5.3736E−01	2.3124E−01	2.7325E−01	1.9779E−01	5.4200E−01	2.8123E−01
F17	Mean	4.9877E+05	1.5315E+06	2.9183E+06	1.1501E+07	3.7307E+04	7.9017E+07	2.3543E+07	1.0507E+09
	Std	3.3574E+05	1.1971E+06	9.9324E+05	2.9081E+06	1.0219E+04	3.3931E+07	1.5184E+07	2.9493E+08
F18	Mean	2.5937E+03	4.3262E+03	1.9498E+03	1.4977E+04	2.0278E+03	2.7039E+09	2.9459E+05	2.2565E+10
	Std	3.2450E+02	1.7094E+03	3.3471E+01	1.0067E+04	2.1925E+01	6.7210E+08	9.5604E+05	4.0705E+09
F19	Mean	1.9255E+03	1.9408E+03	1.9178E+03	1.9432E+03	1.9192E+03	2.4003E+03	2.0586E+03	5.6575E+03
	Std	1.7111E+01	2.2768E+01	2.5989E+00	4.1544E+00	2.0956E+00	9.2196E+01	4.3932E+01	1.2839E+03
F20	Mean	2.7843E+03	5.3820E+03	7.6201E+03	1.2000E+04	2.2434E+03	4.8444E+04	5.7962E+04	5.5550E+05
	Std	7.1792E+02	2.5939E+03	1.9478E+03	3.9044E+03	4.2596E+01	1.3875E+04	1.8990E+04	4.0634E+05
F21	Mean	3.0989E+05	7.4692E+05	1.6829E+06	4.4611E+06	7.2525E+03	1.3752E+07	5.2585E+06	2.0742E+08
	Std	1.3879E+05	1.4451E+06	6.7850E+05	1.6576E+06	8.6239E+02	4.3410E+06	3.6101E+06	9.4565E+07
F22	Mean	2.9087E+03	3.3087E+03	2.8861E+03	2.8979E+03	2.7960E+03	5.0267E+03	3.7516E+03	8.5416E+05
	Std	3.0349E+02	3.4025E+02	1.3306E+02	1.7525E+02	1.2731E+02	2.7959E+02	3.3540E+02	7.4423E+05
F23	Mean	2.5000E+03	2.6440E+03	2.6440E+03	2.5000E+03	2.6441E+03	2.5000E+03	2.5000E+03	2.5015E+03
	Std	2.2452E−08	2.9114E−02	8.6295E−05	6.6937E−04	1.2726E−02	3.8664E−04	0.0000E+00	5.7444E−01
F24	Mean	2.6000E+03	2.6786E+03	2.6603E+03	2.6000E+03	2.6769E+03	2.6000E+03	2.6000E+03	2.6004E+03
	Std	5.1366E−13	4.2888E+00	3.7369E+00	4.5493E−03	2.0714E+00	2.0774E−03	0.0000E+00	1.3536E−01
F25	Mean	2.7000E+03	2.7232E+03	2.7156E+03	2.7000E+03	2.7165E+03	2.7000E+03	2.7000E+03	2.7000E+03
	Std	5.4726E−13	7.1652E+00	1.4428E+00	1.6050E−05	1.4787E+00	8.9987E−06	0.0000E+00	9.4600E−03
F26	Mean	2.7368E+03	2.7805E+03	2.7039E+03	2.7005E+03	2.7004E+03	2.7050E+03	2.7040E+03	2.8000E+03
	Std	4.8878E+01	7.4227E+01	1.8368E+01	6.9352E−02	2.8754E−02	4.1763E−01	1.8142E+01	1.0257E−04
F27	Mean	3.0010E+03	4.1326E+03	3.5633E+03	3.3253E+03	3.5511E+03	2.9000E+03	2.9000E+03	6.2548E+03
	Std	2.1319E+02	1.3759E+02	3.0854E+02	6.1412E+02	4.2828E+02	7.0698E−05	1.3876E−12	7.1969E+02
F28	Mean	3.0000E+03	5.9252E+03	4.2260E+03	3.0345E+03	4.5327E+03	3.0000E+03	3.0000E+03	1.8710E+04
	Std	3.1447E−07	9.3591E+02	7.1001E+01	1.8876E+02	1.0657E+02	8.6107E−04	1.3876E−12	1.8366E+03
F29	Mean	3.1000E+03	1.4373E+07	4.5595E+03	6.2146E+03	8.5455E+03	4.0677E+03	3.1000E+03	4.6542E+07
	Std	2.6081E−03	3.5718E+07	3.0829E+02	2.4175E+03	1.5276E+03	1.1930E+03	0.0000E+00	2.4654E+08
F30	Mean	3.2000E+03	2.9902E+04	1.3395E+04	1.0882E+04	1.5651E+04	3.2403E+03	3.2000E+03	6.5958E+07
	Std	9.6308E−05	1.0889E+04	1.0396E+03	5.1662E+03	1.0830E+03	3.2663E+01	0.0000E+00	1.9283E+07

Table 21 The comparison of the statistical results obtained by all the algorithms for IEEE CEC2014 benchmark set at $D = 100$

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F1	Mean	3.0373E+07	1.6890E+08	1.0391E+08	1.5800E+09	4.5335E+07	3.8793E+09	4.9043E+08	1.2539E+10
	Std	6.2481E+06	8.2566E+07	1.8017E+07	2.7095E+08	7.9694E+06	5.7769E+08	1.3112E+08	2.1241E+09
F2	Mean	1.0115E+04	3.4788E+04	5.9123E+02	2.6788E+04	6.5006E+07	2.2886E+11	4.3842E+10	3.1731E+11
	Std	1.0061E+04	3.3344E+04	5.5123E+02	2.8121E+04	9.9092E+06	1.2638E+10	9.1261E+09	1.4322E+10
F3	Mean	3.8941E+03	4.1767E+03	1.9645E+03	5.0625E+03	2.7773E+03	2.6526E+05	2.4072E+05	3.3420E+05
	Std	1.4962E+03	6.2394E+03	5.9028E+02	1.7422E+03	4.3237E+02	1.0278E+04	2.4386E+04	1.5722E+04
F4	Mean	6.7495E+02	8.5885E+02	6.2133E+02	5.9610E+02	7.6936E+02	5.0318E+04	4.8556E+03	1.1088E+05
	Std	3.6040E+01	1.2037E+02	2.8402E+01	2.8656E+01	2.8265E+01	7.0766E+03	1.1494E+03	1.2470E+04
F5	Mean	5.2000E+02	5.2119E+02	5.2067E+02	5.2126E+02	5.2033E+02	5.2132E+02	5.2073E+02	5.2135E+02
	Std	9.6536E−05	3.9274E−02	4.0364E−02	2.7287E−02	2.3212E−02	2.5498E−02	1.1313E−01	2.9799E−02
F6	Mean	6.2646E+02	6.9565E+02	6.7992E+02	7.3881E+02	6.9428E+02	7.4993E+02	7.3219E+02	7.6331E+02
	Std	4.4928E+00	6.1348E+00	3.8667E+00	2.2096E+00	2.0372E+00	3.1247E+00	9.2966E+00	2.5280E+00
F7	Mean	7.0000E+02	7.0004E+02	7.0000E+02	7.0000E+02	7.0118E+02	2.9078E+03	1.0831E+03	3.9215E+03
	Std	3.0756E−03	1.5114E−01	1.0194E−04	1.5296E−03	2.3155E−02	1.3311E+02	5.6687E+01	1.4485E+02
F8	Mean	9.4185E+02	1.0147E+03	8.0000E+02	1.1964E+03	9.3585E+02	2.1008E+03	1.6015E+03	2.2069E+03
	Std	2.1854E+01	4.0285E+01	1.4162E−13	8.7542E+00	9.1889E+00	1.5963E+01	3.3608E+01	4.3962E+01
F9	Mean	1.1138E+03	1.5713E+03	1.3290E+03	1.7844E+03	1.6009E+03	2.3387E+03	1.8800E+03	2.3926E+03
	Std	3.0262E+01	1.5456E+02	3.3715E+01	3.0506E+01	4.6289E+01	3.2350E+01	4.4718E+01	4.2799E+01
F10	Mean	9.9232E+03	6.6889E+03	1.0015E+03	1.1868E+04	3.6278E+03	3.1069E+04	2.0832E+04	3.2403E+04
	Std	1.3765E+03	9.8903E+02	8.2675E−01	4.9386E+02	2.3908E+02	3.8020E+02	7.7268E+02	7.8972E+02
F11	Mean	1.0804E+04	1.5441E+04	1.3692E+04	2.9800E+04	1.4038E+04	3.0720E+04	1.9658E+04	3.3206E+04
	Std	1.1231E+03	1.1278E+03	8.0169E+02	8.4141E+02	6.7302E+02	5.0284E+02	1.1201E+03	8.3069E+02
F12	Mean	1.2000E+03	1.2022E+03	1.2005E+03	1.2032E+03	1.2005E+03	1.2042E+03	1.2023E+03	1.2045E+03
	Std	1.1272E−02	3.5883E−01	4.5458E−02	2.0633E−01	5.1288E−02	1.6855E−01	6.8393E−01	4.2190E−01
F13	Mean	1.3004E+03	1.3007E+03	1.3005E+03	1.3006E+03	1.3004E+03	1.3080E+03	1.3030E+03	1.3097E+03
	Std	3.6105E−02	8.9022E−02	3.5586E−02	7.1605E−02	3.6641E−02	2.4736E−01	7.2045E−01	2.6337E−01
F14	Mean	1.4003E+03	1.4006E+03	1.4003E+03	1.4004E+03	1.4002E+03	2.0542E+03	1.5150E+03	2.3653E+03
	Std	1.9093E−02	2.8319E−01	1.7061E−02	1.8109E−01	1.5161E−02	4.2125E+01	2.0347E+01	3.8085E+01
F15	Mean	1.5162E+03	1.6103E+03	1.5539E+03	1.5813E+03	1.6432E+03	7.1870E+06	7.7084E+04	3.1953E+07
	Std	2.3681E+00	2.4699E+01	3.3440E+00	2.4362E+00	1.0666E+01	1.4393E+06	4.4472E+04	8.1341E+06
F16	Mean	1.6431E+03	1.6452E+03	1.6408E+03	1.6465E+03	1.6424E+03	1.6466E+03	1.6451E+03	1.6473E+03
	Std	9.6941E−01	3.6811E−01	5.7743E−01	2.1849E−01	5.1940E−01	2.7699E−01	8.9190E−01	3.3166E−01
F17	Mean	1.7872E+06	1.5241E+07	1.5337E+07	1.0572E+08	6.1941E+05	4.5688E+08	6.9203E+07	2.2487E+09
	Std	6.1682E+05	1.0119E+07	3.2867E+06	2.3680E+07	1.9965E+05	9.7137E+07	2.9614E+07	4.2971E+08
F18	Mean	3.0950E+03	1.6532E+05	2.1420E+03	7.5992E+03	2.9321E+03	1.5597E+10	1.2178E+06	4.4608E+10
	Std	3.7091E+02	6.7432E+05	7.2154E+01	1.4856E+04	1.2325E+02	2.3392E+09	6.2212E+06	5.2934E+09
F19	Mean	1.9752E+03	2.0297E+03	1.9860E+03	2.0107E+03	1.9914E+03	4.4807E+03	2.3930E+03	1.3461E+04
	Std	2.3966E+01	2.1378E+01	1.4933E+01	1.5642E+00	1.7086E+01	4.0873E+02	8.9316E+01	1.7666E+03
F20	Mean	3.3312E+03	1.7176E+04	1.2120E+04	3.4767E+04	3.8046E+03	2.7952E+05	1.5113E+05	1.8854E+06
	Std	3.1286E+02	4.0600E+03	3.6516E+03	7.2559E+03	5.0775E+02	5.9664E+04	3.0685E+04	1.1003E+06
F21	Mean	1.6299E+06	5.6538E+06	7.0572E+06	4.1190E+07	6.7877E+04	1.5503E+08	1.7784E+07	5.2156E+08
	Std	6.9097E+05	3.7075E+06	2.0203E+06	8.1843E+06	1.9718E+04	3.9951E+07	8.6440E+06	1.0203E+08
F22	Mean	3.5471E+03	4.9940E+03	4.0170E+03	5.1262E+03	3.8985E+03	1.1893E+04	5.3235E+03	4.0948E+05
	Std	3.6910E+02	4.2864E+02	2.7537E+02	2.2846E+02	2.8120E+02	1.7024E+03	5.7711E+02	2.8636E+05
F23	Mean	2.5000E+03	2.6523E+03	2.6482E+03	2.5000E+03	2.6488E+03	2.5000E+03	2.5000E+03	2.5009E+03
	Std	9.7777E−11	2.7861E+00	9.7546E−03	1.4844E−04	7.4011E−02	3.2680E−04	0.0000E+00	2.5340E−01
F24	Mean	2.6000E+03	2.7912E+03	2.7610E+03	2.6000E+03	2.8005E+03	2.6000E+03	2.6000E+03	2.6004E+03
	Std	2.4628E−07	5.4624E+00	2.0714E+00	4.0190E−03	3.9007E+00	2.7675E−03	0.0000E+00	1.3023E−01
F25	Mean	2.7000E+03	2.7908E+03	2.7572E+03	2.7000E+03	2.7668E+03	2.7000E+03	2.7000E+03	2.7000E+03
	Std	1.4700E−09	1.6993E+01	4.4215E+00	7.6659E−06	3.9150E+00	5.9650E−06	0.0000E+00	7.3391E−03

Table 21 (continued)

Function		CDESSA	ALCPSO	CLPSO	DECLS	WDE	ESSA	CMSSA	CSSA
F26	Mean	2.8000E+03	2.8135E+03	2.7959E+03	2.7373E+03	2.7398E+03	2.7579E+03	2.7967E+03	2.8000E+03
	Std	2.5333E−13	6.2806E+01	3.2300E+01	4.8494E+01	4.9461E+01	4.5940E+01	1.8157E+01	4.8990E−05
F27	Mean	2.9000E+03	5.6893E+03	4.9494E+03	3.4481E+03	5.1651E+03	2.9000E+03	2.9000E+03	1.0792E+04
	Std	2.7241E−05	1.9037E+02	3.1812E+02	1.2474E+03	6.8814E+02	3.8811E−05	2.3126E−12	1.1237E+03
F28	Mean	3.0000E+03	9.7895E+03	6.7116E+03	3.0000E+03	8.0358E+03	3.0000E+03	3.0000E+03	4.1848E+04
	Std	2.1634E−08	1.4031E+03	7.5170E+02	1.7129E−01	5.6013E+02	4.4692E−04	2.3126E−12	3.3839E+03
F29	Mean	3.1000E+03	8.1801E+07	6.4705E+03	6.6327E+03	1.3394E+04	3.9823E+03	3.1000E+03	1.5244E+06
	Std	6.2494E−04	1.4183E+08	8.1234E+02	6.6160E+02	1.6613E+03	8.0442E+02	0.0000E+00	1.0125E+06
F30	Mean	3.2000E+03	2.8557E+05	3.9946E+04	2.1366E+04	1.8428E+04	3.3531E+03	3.2000E+03	4.8488E+08
	Std	1.1902E−04	2.0854E+05	9.8120E+03	2.6105E+04	1.4322E+03	1.6098E+02	0.0000E+00	2.0860E+08

Table 22 The statistics results of all the methods on the fitness value

Dataset	BMFO		BSSA		BWOA		BFOA		SSAPSO		CDESSA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Breastcancer	0.0309	0.0010	0.0322	0.0017	0.0319	0.0020	0.0316	0.0011	0.0355	0.0026	0.0295	0.0023
BreastEW	0.0395	0.0026	0.0418	0.0013	0.0390	0.0022	0.0384	0.0021	0.0515	0.0026	0.0387	0.0020
Exactly	0.0336	0.0066	0.0342	0.0050	0.0319	0.0039	0.0414	0.0056	0.0454	0.0068	0.0324	0.0046
Primary-tumor	0.5575	0.0079	0.5563	0.0081	0.5577	0.0065	0.5582	0.0057	0.5488	0.0082	0.5449	0.0041
Heart	0.0983	0.0057	0.0981	0.0054	0.1004	0.0078	0.1000	0.0070	0.1146	0.0079	0.0922	0.0062
M-of-n	0.0306	0.0014	0.0316	0.0024	0.0285	0.0017	0.0330	0.0032	0.0386	0.0020	0.0291	0.0013
SpectEW	0.1043	0.0076	0.1050	0.0052	0.0991	0.0059	0.1064	0.0068	0.1212	0.0041	0.0975	0.0062
CongressEW	0.0301	0.0032	0.0315	0.0022	0.0308	0.0028	0.0328	0.0031	0.0417	0.0036	0.0317	0.0033
CTG3	0.0766	0.0015	0.0766	0.0026	0.0760	0.0028	0.0774	0.0016	0.0849	0.0015	0.0768	0.0017
Cleveland_heart	0.0987	0.0075	0.1044	0.0071	0.1058	0.0081	0.1086	0.0037	0.1179	0.0069	0.1044	0.0064
Brain_Tumor2	0.1134	0.0148	0.1183	0.0255	0.0998	0.0174	0.1231	0.0219	0.1554	0.0249	0.1215	0.0181
Tumors_14	0.3096	0.0074	0.3114	0.0094	0.3148	0.0120	0.3160	0.0106	0.3365	0.0098	0.3108	0.0056

Table 23 The statistics results of all the methods on the error value

Dataset	BMFO		BSSA		BWOA		BFOA		SSAPSO		CDESSA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Breastcancer	0.0114	0.0014	0.0115	0.0019	0.0115	0.0024	0.0128	0.0021	0.0093	0.0025	0.0107	0.0018
BreastEW	0.0167	0.0033	0.0193	0.0018	0.0158	0.0020	0.0158	0.0029	0.0220	0.0025	0.0159	0.0029
Exactly	0.0050	0.0060	0.0055	0.0044	0.0030	0.0034	0.0120	0.0052	0.0150	0.0066	0.0045	0.0047
Primary-tumor	0.5554	0.0089	0.5520	0.0086	0.5547	0.0077	0.5543	0.0070	0.5391	0.0091	0.5369	0.0047
Heart	0.0741	0.0058	0.0796	0.0063	0.0778	0.0087	0.0778	0.0078	0.0889	0.0102	0.0685	0.0072
M-of-n	0.0020	0.0014	0.0020	0.0020	0.0000	0.0014	0.0030	0.0027	0.0070	0.0019	0.0010	0.0008
SpectEW	0.0862	0.0081	0.0863	0.0060	0.0823	0.0069	0.0877	0.0071	0.0974	0.0041	0.0802	0.0062
CongressEW	0.0127	0.0035	0.0115	0.0020	0.0116	0.0023	0.0137	0.0026	0.0161	0.0034	0.0115	0.0031
CTG3	0.0574	0.0013	0.0574	0.0023	0.0576	0.0021	0.0571	0.0017	0.0600	0.0017	0.0576	0.0021
Cleveland_heart	0.0779	0.0070	0.0840	0.0080	0.0822	0.0083	0.0892	0.0047	0.0921	0.0082	0.0842	0.0080
Brain_Tumor2	0.0925	0.0157	0.1008	0.0276	0.0808	0.0182	0.1027	0.0232	0.1342	0.0268	0.1013	0.0193
Tumors_14	0.2975	0.0078	0.3006	0.0102	0.3034	0.0129	0.3044	0.0112	0.3183	0.0107	0.2975	0.0061

Table 24 The statistics results of all the methods on the feature number

Dataset	BMFO		BSSA		BWOA		BFOA		SSAPSO		CDESSA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Breastcancer	3.6000	0.1889	3.7000	0.2718	3.7000	0.2530	3.6000	0.3062	4.6500	0.2470	3.6489	0.3665
BreastEW	13.7500	0.5701	14.2500	0.7714	14.1500	0.6125	14.4000	0.7043	18.2500	0.7608	14.0703	0.9146
Exactly	7.4500	0.2757	7.4500	0.2914	7.4000	0.2916	7.8000	0.1889	8.1000	0.1792	7.3000	0.2091
Primary-tumor	10.1000	0.3779	10.7500	0.5379	10.6500	0.5350	10.6000	0.4522	12.3000	0.6574	9.9267	0.5356
Heart	6.7500	0.5125	6.5500	0.4358	6.8500	0.3706	6.7000	0.1829	7.8000	0.5317	6.4246	0.5628
M-of-n	7.4500	0.1350	7.6000	0.1792	7.2500	0.2300	7.6500	0.2633	8.2500	0.1338	7.4000	0.2714
SpectEW	9.3500	0.8386	9.4500	0.6579	9.2000	0.6964	9.5000	0.5990	12.8000	0.5658	9.7897	0.4318
CongressEW	6.1000	0.7543	6.3500	0.7554	6.1000	0.5012	6.2500	0.3496	8.6000	0.5334	6.0750	0.5534
CTG3	9.1000	0.4761	8.8500	0.3808	9.0000	0.5793	9.8000	0.4990	11.9000	0.5724	8.9000	0.4977
Cleveland_heart	6.5000	0.4624	6.3500	0.4625	6.8000	0.8155	6.5500	0.3802	7.8500	0.6147	6.2362	0.4794
Brain_Tumor2	5305.3000	40.7346	4543.5500	178.7260	4795.7000	99.1515	5271.3500	63.6775	5482.7500	244.8378	5216.6500	71.2003
Tumors_14	8114.2500	56.9465	7792.2000	152.9663	7858.0000	178.5936	8101.9500	58.0884	10,524.3000	376.6380	7960.4500	92.1092

Acknowledgements This work was supported in part by the college-enterprise cooperation project of the domestic visiting engineer of colleges (FG2020077), Zhejiang, China, General research project of Zhejiang Provincial Education Department (Y201942618), Zhejiang, China, the National Natural Science Foundation of China (62076185, U1809209), the Beijing Natural Science Foundation (L182015), Zhejiang Provincial Natural Science Foundation of China (LY21F020030), Wenzhou Science & Technology Bureau (2018ZG016). We acknowledge comments of the reviewers.

References

1. Chen H, Zhang Q, Luo J, Xu Y, Zhang X (2019) An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105884
2. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028
3. Yu H, Zhao N, Wang P, Chen H, Li C (2020) Chaos-enhanced synchronized bat optimizer. Appl Math Model 77:1201–1215. https://doi.org/10.1016/j.apm.2019.09.029
4. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323. https://doi.org/10.1016/j.future.2020.03.055
5. Wei Y, Lv H, Chen M, Wang M, Heidari AA, Chen H, Li C (2020) Predicting entrepreneurial intention of students: an extreme learning machine with gaussian barebone Harris Hawks optimizer. IEEE Access 8:76841–76855. https://doi.org/10.1109/ACCESS.2020.2982796
6. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
7. Zong WG, Joong HK, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. SIMULATION 76:60–68. https://doi.org/10.1177/003754970107600201
8. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95—international conference on neural networks, vol 1944, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
9. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
10. Yang X-S (2012) Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation, Berlin, pp 240–249. https://doi.org/10.1007/978-3-642-32894-7_27
11. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8:256–279. https://doi.org/10.1109/TEVC.2004.826067
12. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11:712–731. https://doi.org/10.1109/TEVC.2007.892759
13. Abbassi A, Abbassi R, Heidari AA, Oliva D, Chen H, Habib A, Jemli M, Wang M (2020) Parameters identification of photovoltaic cell models using enhanced exploratory salp chains-based approach. Energy 198:117333. https://doi.org/10.1016/j.energy.2020.117333
14. Zhang Q, Chen H, Heidari AA, Zhao X, Xu Y, Wang P, Li Y, Li C (2019) Chaos-induced and mutation-driven schemes boosting

salp chains-inspired optimizers. IEEE Access 7:31243–31261. https://doi.org/10.1109/access.2019.2902306

15. Gupta S, Deep K, Heidari AA, Moayedi H, Chen H (2021) Harmonized salp chain-built optimization. Eng Comput 37:1049–1079. https://doi.org/10.1007/s00366-019-00871-5
16. Faris H, Mirjalili S, Aljarah I, Mafarja M, Heidari AA (2020) Nature-inspired optimizers: theories, literature reviews and applications. Springer International Publishing, Cham, pp 185–199. https://doi.org/10.1007/978-3-030-12127-3_11
17. Faris H, Mafarja MM, Heidari AA, Aljarah I, Al-Zoubi AM, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl Based Syst 154:43–67. https://doi.org/10.1016/j.knosys.2018.05.009
18. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. Appl Intell 48:3462–3481. https://doi.org/10.1007/s10489-018-1158-6
19. Tubishat M, Ja'afar S, Alswaitti M, Mirjalili S, Idris N, Ismail MA, Omar MS (2021) Dynamic Salp swarm algorithm for feature selection. Expert Syst Appl 164:113873. https://doi.org/10.1016/j.eswa.2020.113873
20. El-Fergany AA (2018) Extracting optimal parameters of PEM fuel cells using salp swarm optimizer. Renew Energy 119:641–648. https://doi.org/10.1016/j.renene.2017.12.051
21. Zhang J, Wang Z, Luo X (2018) Parameter estimation for soil water retention curve using the salp swarm algorithm. Water 10(6):815. https://doi.org/10.3390/w10060815
22. Hussien AG, Hassanien AE, Houssein EH (2017) Swarming behaviour of salps algorithm for predicting chemical compound activities. In: 2017 eighth international conference on intelligent computing and information systems (ICICIS), pp 315–320. https://doi.org/10.1109/INTELCIS.2017.8260072
23. Zhao H, Huang G, Yan N (2018) Forecasting energy-related CO_2 emissions employing a novel SSA-LSSVM model: considering structural factors in China. Energies. https://doi.org/10.3390/en11040781
24. Ateya AA, Muthanna A, Vybornova A, Algarni AD, Abuarqoub A, Koucheryavy Y, Koucheryavy A (2019) Chaotic salp swarm algorithm for SDN multi-controller networks. Eng Sci Technol 22:1001–1012. https://doi.org/10.1016/j.jestch.2018.12.015
25. Yang B, Zhong L, Zhang X, Shu H, Yu T, Li H, Jiang L, Sun L (2019) Novel bio-inspired memetic salp swarm algorithm and application to MPPT for PV systems considering partial shading condition. J Clean Prod 215:1203–1222. https://doi.org/10.1016/j.jclepro.2019.01.150
26. Wang J, Gao Y, Chen X (2018) A novel hybrid interval prediction approach based on modified lower upper bound estimation in combination with multi-objective salp swarm algorithm for short-term load forecasting. Energies 11:1–30. https://doi.org/10.3390/en11061561
27. Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. Energy Convers Manag 179:362–372. https://doi.org/10.1016/j.enconman.2018.10.069
28. Abadi MQH, Rahmati S, Sharifi A, Ahmadi M (2021) HSSAGA: designation and scheduling of nurses for taking care of COVID-19 patients using novel method of hybrid salp swarm algorithm and genetic algorithm. Appl Soft Comput 108:107449. https://doi.org/10.1016/j.asoc.2021.107449
29. Abd el-sattar S, Kamel S, Ebeed M, Jurado F (2021) An improved version of salp swarm algorithm for solving optimal power flow problem. Soft Comput 25:4027–4052. https://doi.org/10.1007/s00500-020-05431-4
30. Ewees AA, Al-qaness MAA, Abd EM (2021) Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times. Appl Math Model 94:285–305. https://doi.org/10.1016/j.apm.2021.01.017
31. Salgotra R, Singh U, Singh S, Singh G, Mittal N (2021) Self-adaptive salp swarm algorithm for engineering optimization problems. Appl Math Model 89:188–207. https://doi.org/10.1016/j.apm.2020.08.014
32. Ibrahim RA, Ewees AA, Oliva D, Abd EM, Lu S (2019) Improved salp swarm algorithm based on particle swarm optimization for feature selection. J Ambient Intell Humaniz Comput 10:3155–3169. https://doi.org/10.1007/s12652-018-1031-9
33. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran M (2019) A new binary salp swarm algorithm: development and application for optimization tasks. Neural Comput Appl 31:1641–1663. https://doi.org/10.1007/s00521-018-3613-z
34. Bairathi D, Gopalani D (2021) An improved salp swarm algorithm for complex multi-modal problems. Soft Comput 25:10441–10465. https://doi.org/10.1007/s00500-021-05757-7
35. Braik M, Sheta A, Turabieh H, Alhiary H (2021) A novel lifetime scheme for enhancing the convergence performance of salp swarm algorithm. Soft Comput 25:181–206. https://doi.org/10.1007/s00500-020-05130-0
36. Nautiyal B, Prakash R, Vimal V, Liang G, Chen H (2021) Improved salp swarm algorithm with mutation schemes for solving global optimization and engineering problems. Eng Comput. https://doi.org/10.1007/s00366-020-01252-z
37. Ouaar F, Boudjemaa R (2021) Modified salp swarm algorithm for global optimisation. Neural Comput Appl 33:8709–8734. https://doi.org/10.1007/s00521-020-05621-z
38. Panda N, Majhi SK (2021) Oppositional salp swarm algorithm with mutation operator for global optimization and application in training higher order neural networks. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-10304-x
39. Ren H, Li J, Chen H, Li C (2021) Stability of salp swarm algorithm with random replacement and double adaptive weighting. Appl Math Model 95:503–523. https://doi.org/10.1016/j.apm.2021.02.002
40. Saafan MM, El-Gendy EM (2021) IWOSSA: an improved whale optimization salp swarm algorithm for solving optimization problems. Expert Syst Appl 176:114901. https://doi.org/10.1016/j.eswa.2021.114901
41. Chen W, Zhang J, Lin Y, Chen N, Zhan Z, Chung HS, Li Y, Shi Y (2013) Particle swarm optimization with an aging leader and challengers. IEEE Trans Evol Comput 17:241–258. https://doi.org/10.1109/TEVC.2011.2173577
42. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10:281–295. https://doi.org/10.1109/TEVC.2005.857610
43. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13:398–417. https://doi.org/10.1109/TEVC.2008.927706
44. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657. https://doi.org/10.1109/TEVC.2006.872133
45. Zhu A, Xu C, Li Z, Wu J, Liu Z (2015) Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. J Syst Eng Electron 26:317–328. https://doi.org/10.1109/JSEE.2015.00037
46. Luo J, Chen H, Zhang Q, Xu Y, Huang H, Zhao X (2018) An improved grasshopper optimization algorithm with application to financial stress prediction. Appl Math Model 64:654–668. https://doi.org/10.1016/j.apm.2018.07.044

47. Çelik E, Öztürk N, Arya Y (2021) Advancement of the search process of salp swarm algorithm for global optimization problems. Expert Syst Appl 182:115292. https://doi.org/10.1016/j.eswa.2021.115292
48. Liu Y, Shi Y, Chen H, Heidari AA, Gui W, Wang M, Chen H, Li C (2021) Chaos-assisted multi-population salp swarm algorithms: framework and case studies. Expert Syst Appl 168:114369. https://doi.org/10.1016/j.eswa.2020.114369
49. Salgotra R, Singh U, Singh G, Singh S, Gandomi AH (2021) Application of mutation operators to salp swarm algorithm. Expert Syst Appl 169:114368. https://doi.org/10.1016/j.eswa.2020.114368
50. Zhang H, Wang Z, Chen W, Heidari AA, Wang M, Zhao X, Liang G, Chen H, Zhang X (2021) Ensemble mutation-driven salp swarm algorithm with restart mechanism: framework and fundamental analysis. Expert Syst Appl 165:113897. https://doi.org/10.1016/j.eswa.2020.113897
51. Andersen V, Nival P (1986) A model of the population dynamics of salps in coastal waters of the Ligurian Sea. J Plankton Res 8:1091–1110. https://doi.org/10.1093/plankt/8.6.1091
52. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99. https://doi.org/10.1016/j.engappai.2006.03.003
53. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579. https://doi.org/10.1016/j.amc.2006.11.033
54. Gao W-F, Huang L-L, Wang J, Liu S-Y, Qin C-D (2016) Enhanced artificial bee colony algorithm through differential evolution. Appl Soft Comput 48:137–150. https://doi.org/10.1016/j.asoc.2015.10.070
55. Xiang W-L, Li Y-Z, Meng X-L, Zhang C-M, An M-Q (2017) A grey artificial bee colony algorithm. Appl Soft Comput 60:1–17. https://doi.org/10.1016/j.asoc.2017.06.015
56. Tian D, Zhao X, Shi Z (2019) Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm Evol Comput 51:100573. https://doi.org/10.1016/j.swevo.2019.100573
57. Wang X, Wang Z, Weng J, Wen C, Chen H, Wang X (2018) A new effective machine learning framework for sepsis diagnosis. IEEE Access 6:48300–48310. https://doi.org/10.1109/ACCESS.2018.2867728
58. Zhang Q, Chen H, Luo J, Xu Y, Wu C, Li C (2018) Chaos enhanced bacterial foraging optimization for global optimization. IEEE Access 6:64905–64919. https://doi.org/10.1109/ACCESS.2018.2876996
59. Luo J, Chen H, Heidari AA, Xu Y, Zhang Q, Li C (2019) Multi-strategy boosted mutative whale-inspired optimization approaches. Appl Math Model 73:109–123. https://doi.org/10.1016/j.apm.2019.03.046
60. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: international conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), pp 695–701. https://doi.org/10.1109/CIMCA.2005.1631345
61. Chen H, Wang M, Zhao X (2020) A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Appl Math Comput 369:124872. https://doi.org/10.1016/j.amc.2019.124872
62. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002
63. Jia D, Zheng G, Khurram KM (2011) An effective memetic differential evolution algorithm based on chaotic local search. Inf Sci 181:3175–3187. https://doi.org/10.1016/j.ins.2011.03.018
64. Civicioglu P, Besdok E, Gunen MA, Atasever UH (2020) Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. Neural Comput Appl 32:3923–3937. https://doi.org/10.1007/s00521-018-3822-5
65. Qais MH, Hasanien HM, Alghuwainem S (2019) Enhanced salp swarm algorithm: application to variable speed wind generators. Eng Appl Artif Intell 80:82–96. https://doi.org/10.1016/j.engappai.2019.01.011
66. Frank AA (2010) UCI machine learning repository
67. Pan W-T (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl Based Syst 26:69–74. https://doi.org/10.1016/j.knosys.2011.07.001
68. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
69. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
70. Yang, S., Yu, X., Ding, M., He, L., Cao, G., Zhao, L.,... Ren, N. (2021). Simulating a combined lysis-cryptic and biological nitrogen removal system treating domestic wastewater at low C/N ratios using artificial neural network. Water research (Oxford), 189, 116576. doi: 10.1016/j.watres.2020.116576
71. Che, H., & Wang, J. (2021). A Two-Timescale Duplex Neurodynamic Approach to Mixed-Integer Optimization. IEEE transaction on neural networks and learning systems, 32(1), 36-48. doi: 10.1109/TNNLS.2020.2973760
72. Meng, Q., Lai, X., Yan, Z., Su, C., & Wu, M. (2021). Motion Planning and Adaptive Neural Tracking Control of an Uncertain Two-Link Rigid-Flexible Manipulator With Vibration Amplitude Constraint. IEEE transaction on neural networks and learning systems, PP, 1-15. doi: 10.1109/TNNLS.2021.3054611
73. Zhang, M., Chen, Y., & Susilo, W. (2020). PPO-CPQ: A Privacy-Preserving Optimization of Clinical Pathway Query for E-Healthcare Systems. IEEE internet of things journal, 7(10), 10660-10672. doi: 10.1109/JIOT.2020.3007518
74. Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part II: numerical results. Int J Numer Methods Eng 21:1601–1623. https://doi.org/10.1002/nme.1620210905
75. Coello Coello CA, Mezura Montes E (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203. https://doi.org/10.1016/S1474-0346(02)00011-3
76. Arora JS (2017) Introduction to optimum design, 4th edn. Academic Press, Boston, pp 601–680. https://doi.org/10.1016/B978-0-12-800806-5.00014-7
77. Krohling RA, Coelho L (2006) Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern Part B (Cybernetics) 36:1407–1416. https://doi.org/10.1109/TSMCB.2006.873185
78. Zahara E, Kao Y-T (2009) Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 36:3880–3886. https://doi.org/10.1016/j.eswa.2008.02.039
79. Li LJ, Huang ZB, Liu F, Wu QH (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85:340–349. https://doi.org/10.1016/j.compstruc.2006.11.020

80. Zhang HL, Cai ZN, Ye XJ, Wang MJ, Kuang FJ, Chen HL, Li CY, Li YP (2020) A multi-strategy enhanced salp swarm algorithm for global optimization. Eng Comput. https://doi.org/10.1007/s00366-020-01099-4
81. Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H (2014) Chaotic Krill Herd algorithm. Inf Sci 274:17–34. https://doi.org/10.1016/j.ins.2014.02.123
82. Coello-Coello CA, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36:219–236. https://doi.org/10.1080/03052150410001647966
83. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
84. Kannan BK, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411. https://doi.org/10.1115/1.2919393
85. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223–229. https://doi.org/10.1115/1.2912596
86. Huang F-Z, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356. https://doi.org/10.1016/j.amc.2006.07.105
87. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422. https://doi.org/10.1016/j.amc.2006.07.134
88. Coelho L (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683. https://doi.org/10.1016/j.eswa.2009.06.044
89. Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7:386–396. https://doi.org/10.1109/TEVC.2003.814902
90. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
91. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33:735–748. https://doi.org/10.1080/03052150108940941
92. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178:3043–3074. https://doi.org/10.1016/j.ins.2008.02.014
93. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10:629–640. https://doi.org/10.1016/j.asoc.2009.08.031
94. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35. https://doi.org/10.1007/s00366-011-0241-y
95. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 40:3951–3978. https://doi.org/10.1016/j.apm.2015.10.040
96. Adarsh BR, Raghunathan T, Jayabarathi T, Yang X-S (2016) Economic dispatch using chaotic bat algorithm. Energy 96:666–675. https://doi.org/10.1016/j.energy.2015.12.096
97. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43:303–315. https://doi.org/10.1016/j.cad.2010.12.015
98. Chen, S., Zhang, J., Meng, F., & Wang, D. (2021). A Markov Chain Position Prediction Model Based on Multidimensional Correction. Complexity (New York, N.Y.), 2021. https://doi.org/10.1155/2021/6677132
99. He, Y., Dai, L., & Zhang, H. (2020). Multi-Branch Deep Residual Learning for Clustering and Beamforming in User-Centric Network. IEEE communications letters, 24(10), 2221-2225. https://doi.org/10.1109/LCOMM.2020.3005947
100. Wu, X., Zheng, W., Chen, X., Zhao, Y., Yu, T., & Mu, D. (2021). Improving high-impact bug report prediction with combination of interactive machine learning and active learning. Information and Software Technology, 133, 106530.
101. Wu, Z., Cao, J., Wang, Y., Wang, Y., Zhang, L., & Wu, J. (2018). hPSD: a hybrid PU-learning-based spammer detection model for product reviews. IEEE transactions on cybernetics, 50(4), 1595-1606.
102. Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. Neural computing and applications, 31(7), 1995-2014.
103. Wang, G. G., Deb, S., & Coelho, L. D. S. (2018). Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. International journal of bio-inspired computation, 12(1), 1-22.
104. Wang, G. G., Deb, S., & Coelho, L. D. S. (2015, December). Elephant herding optimization. In 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI) (pp. 1-5). IEEE. doi: 10.1109/ISCBI.2015.8
105. Wang, G. G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Computing, 10(2), 151-164.
106. Liu, X. F., Zhan, Z. H., Lin, Y., Chen, W. N., Gong, Y. J., Gu, T. L., ... & Zhang, J. (2018). Historical and heuristic-based adaptive differential evolution. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(12), 2623-2635.
107. Zhao, H., Zhan, Z. H., Lin, Y., Chen, X., Luo, X. N., Zhang, J., ... & Zhang, J. (2019). Local binary pattern-based adaptive differential evolution for multimodal optimization problems. IEEE transactions on cybernetics, 50(7), 3343-3357.
108. Zhan, Z. H., Wang, Z. J., Jin, H., & Zhang, J. (2019). Adaptive distributed differential evolution. IEEE transactions on cybernetics, 50(11), 4633-4647.
109. Zhan, Z. H., Liu, X. F., Zhang, H., Yu, Z., Weng, J., Li, Y., ... & Zhang, J. (2016). Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version. IEEE Transactions on Parallel and Distributed Systems, 28(3), 704-716.
110. Liu, X. F., Zhan, Z. H., & Zhang, J. (2021) Resource-Aware Distributed Differential Evolution for Training Expensive Neural-Network-Based Controller in Power Electronic Circuit. IEEE Transactions on Neural Networks and Learning Systems 1-11 10.1109/TNNLS.2021.3075205
111. Chen, Z. G., Zhan, Z. H., Wang, H., & Zhang, J. (2019). Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems. IEEE Transactions on Evolutionary Computation, 24(4), 708-719.
112. Chen, H., Li, S., Heidari, A. A., Wang, P., Li, J., Yang, Y., ... & Huang, C. (2020). Efficient multi-population outpost fruit fly-driven optimizers: Framework and advances in support vector machines. Expert Systems with Applications, 142, 112999.
113. Chen, H., Yang, C., Heidari, A. A., & Zhao, X. (2020). An efficient double adaptive random spare reinforced whale optimization algorithm. Expert Systems with Applications, 154, 113018.
114. Zhang, H., Heidari, A. A., Wang, M., Zhang, L., Chen, H., & Li, C. (2020). Orthogonal Nelder-Mead moth flame method for parameters identification of photovoltaic modules. Energy Conversion and Management, 211, 112764.
115. Ridha, H. M., Heidari, A. A., Wang, M., & Chen, H. (2020). Boosted mutation-based Harris hawks optimizer for parameters

identification of single-diode solar cell models. Energy Conversion and Management, 209, 112660.

116. Chen, H., Jiao, S., Wang, M., Heidari, A. A., & Zhao, X. (2020). Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. Journal of Cleaner Production, 244, 118778.

117. Chen, H., Heidari, A. A., Chen, H., Wang, M., Pan, Z., & Gandomi, A. H. (2020). Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. Future Generation Computer Systems, 111, 175-198.