Applied Soft Computing 115 (2022) 108126

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An intelligent chaotic clonal optimizer

Vahideh Sahargahi, Vahid Majidnezhad *, Saeid Taghavi Afshord, Yasser Jafari

Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran

A R T I C L E I N F O

Article history:
Received 16 February 2021
Received in revised form 13 October 2021
Accepted 18 October 2021
Available online 27 November 2021

Keywords:
Continuous Optimization problems
Metaheuristic
Intelligent Clonal Optimizer
Conservative selection operator
Clonal Operator

A B S T R A C T

Increasing complexity of real world problems motivated an area to explore efficient optimization methods to solve such problems. Existing optimization algorithms cannot solve all type of problems efficiently (NFL theorem), so new algorithms are proposed to find the better solutions for such complex optimization problems. However, their efficiency and performance can be still improved. Therefore, to follow this vital purpose, in this paper, a novel metaheuristic algorithm, called intelligent clonal optimizer (ICO), is proposed to solve continuous optimization problems. In the proposed algorithm, the initial population is generated through the chaos theory to enhance its exploration capability. It lacks any crossover operator. Instead, a novel clonal operator copying candidate solutions according to their fitness in a self-adaptive way is proposed. Cloning each parent is carried out by two methods, and according to these methods, each offspring is located near the parent or in direction of temporary target. The offsprings are classified to two classes. In addition, a novel conservative selection operator is proposed. According to this operator, the new population is selected from two classes of offsprings and current population by maintaining population diversity. The performance of the ICO algorithm is assessed on 39 well-known unimodal, multimodal, fixed-dimensional multimodal, composite and CEC2019 benchmark functions as well as three engineering application problems. Results of the proposed ICO are compared to sixteen state-of-art metaheuristic algorithms in three categories including the most well-known and recently developed algorithms and the best performer of IEEE CEC competitions using statistical analysis, scalability analysis, Wilcoxon Signed-Rank Test, Friedman test, computational time analysis and convergence analysis. The obtained results proved that ICO performs better than state-of-art metaheuristics in sense of scalability and accurate convergence. According to average rank of Friedman test, the proposed ICO is firstly ranked among others.

1. Introduction

Optimization problems involve finding the best solutions from a large set of solutions. Such problems are categorized as different classes such as constrained or unconstrained, discrete or continuous, static or dynamic, and single-objective or multi-objective. Since most of real life optimization problems have high complexity, nonlinear constraints, mutual dependency of variables and a wide range of solutions, finding some solutions for such problems can be a challenging task. Therefore, many researchers proposed optimization methods typically classified as mathematical programming approaches and metaheuristic methods to solve these problems.

Mathematical optimization methods are divided into two categories including deterministic and stochastic [1] approaches. In deterministic approaches, analytical properties of the problem are considered to generate a sequence of points converging to a global optimal solution. Although these methods are efficient for problems with linear search spaces (unimodal), they are less efficient and less effective tools to solve multi-modal, highly complex, and high-dimensional optimization problems because they are prone to local optima entrapment. Like meta-heuristic algorithms, stochastic methods generate and use random variables. In order to find the global or near global optimal results, these algorithms are used to search the domain globally.

Advantages of metaheuristics include simplicity, derivative free, flexibility and local optima avoidance [2]. Different types of metaheuristic algorithms are classified as evolutionary algorithms, physics-based, swarm intelligence, and human based algorithms [1]. Table 1 shows some of the famous optimization algorithms of last 10 years.

Evolutionary algorithms integrate aspects of natural choice with continuity of coordination. An evolutionary algorithm (EA) protects the population of structures in terms of selection rules, reconfiguration, change and survival. These structures are based on genetic operations. In this method, the environment determines the coordination or performance of each population member, and more coordinated members are employed for regeneration. Different types of evolutionary algorithms are Genetic

* Corresponding author.
E-mail address: Vahidmn@iaushab.ac.ir (V. Majidnezhad).

https://doi.org/10.1016/j.asoc.2021.108126
1568-4946/

Table 1
Optimization algorithms.

Algorithm	Year
Firefly Algorithm (FA) [3]	2010
TLBO (Teaching–Learning-Based Optimization) [4]	2011
KH (Krill Herd) [5]	2012
DE (Dolphin Echolocation) [6]	2013
BMO (Bird Mating Optimizer) [7]	2014
GWO (Gray Wolf Optimization) [8]	2014
ALO (Ant Lion Optimizer) [9]	2015
MFO (Moth–Flame Optimization) [10]	2015
DA (Dragonfly Algorithm) [11]	2016
MVO (Multi-Verse Optimizer) [12]	2016
WOA (Whale Optimization Algorithm) [13]	2016
MOGWO (Multi-Objective Grey Wolf Optimizer) [14]	2016
LOA (Lion Optimization Algorithm) [15])	2016
WEO (Water Evaporation Optimization) [16]	2016
SBO (Satin Bowerbird Optimizer) [17]	2017
GOA (Grasshopper Optimization Algorithm) [18]	2017
SHO (Selfish Herds optimizer) [19]	2017
SSA(Salp Swarm Algorithm) [20]	2017
SELO (Socio Evolution & Learning Optimization Algorithm [21])	2018
RE-GWO (Random Walk Grey Wolf Optimizer) [22]	2019
HHO (Harris Hawks Optimizer) [23]	2019
SFO (Sail Fish Optimizer) (Shadravan et al. 2019)	2019
SMO(Social Mimic Optimization Algorithm) [24]	2019
PRO(Poor and rich optimization algorithm) [25]	2019
SASS (Spherical search optimization algorithm) [2]	2019
EO(Equilibrium optimizer) [1]	2020
MSCA (Multi-strategy enhanced Sine Cosine Algorithm) [26]	2020
LFSHO (Selfish herd optimizer with levy-flight distribution) [27]	2020
DDSCA (Dimension Dynamic Sine Cosine Algorithm) [28]	2020
OBCWOA (WOA with chaos mechanism based on	2020
quasi-opposition) [29]	2020
CMFPA (flower pollination algorithm) [30]	2021
TriTSA (Triple Tree-Seed Algorithm) [31]	2021
BISA (Bayesian Interactive Search Algorithm) [32]	2021
HFA-GA (Hybrid firefly algorithm with grouping attraction) [33]	2021
CSA (Cooperation search algorithm) [34]	

Algorithm (GA) [35], Genetic Programming Algorithm (GP) [36], Evolution Strategies (ES) [37], Invasive Weed Optimization Algorithm (IWO) [38], differential Evolution Algorithm (DE) [39], and differential Evolution Mapping Techniques (BMV) [40].

Simulated Annealing algorithm (SA) [41], Gravitational Search Algorithm (GSA) [42] and Equilibrium Optimizer (EO) [1] are three instances of physical structure simulation algorithms.

Some instances of the swarm intelligence (SI) algorithms are Particle Swarm Optimization (PSO) [43], Ameliorated Particle Swarm Optimizer (A-PSO) [44], Grey Wolf Optimizer (GWO) [8], Hybrid Bat Algorithm (HBA) [45], Ant Colony Optimization (ACO) [46], and ABC with Adaptive Heterogeneous Competition (ABC-AHC) [47], New Caledonian (NC) Crow Learning Algorithm (NC-CLA) [48] algorithms.

Human behavior is used as the main idea of some optimization algorithms named human based algorithms. Some instances involve Tabu Search (TS) [49], Imperialist Competitive Algorithm (ICA) [50], Colliding Bodies Optimization (CBO) [51], Mine Blast Algorithm (MBA) [52], Group Counseling Optimization (GCO) [53] and Adolescent Identity Search Algorithm (AISA) [54].

Arithmetic Optimization Algorithm (AOA) [55], Red Fox optimization algorithm (RFA) [56], Chameleon Swarm Algorithm (CSA) [57], Flow Direction Algorithm (FDA) [58] and Sheep Flock Optimization Algorithm (SFOA) [59] are among algorithms that have been introduced in the last year. Polap and Wozniak presented mathematical model of red fox habits, food search, and hunting and population development when escape from predators [56]. The descriptive model is based on local and global optimization methods with a clone mechanism. The experimental results are compared with other meta-heuristic algorithms in order to show the advantages of this method in newly developed model. A new meta-heuristic algorithm called Chameleon Swarm Algorithm (CSA) in solving general numerical optimization problems is suggested by Braik [57]. CSA inspired by dynamic behavior of crows when exploring and hunting food on trees, deserts and near swamps. According to results, CSA offers a solution that is desirable or close to global solution with better performance than other meta-heuristic methods. Abualigah et al. proposed a new meta-heuristic method called Arithmetic Optimization Algorithm (AOA) which uses the distribution behavior of the main arithmetic operators in mathematics [55]. Experimental results show that AOA provides very promising results in solving challenging optimization problems compared to other well-known optimization algorithms. Flow Direction Algorithm (FDA) is proposed in paper [58]. The algorithm mimics direction of flow to output point with the lowest height in drainage area. In other words, flow moves to neighbor with minimum high target function or the best target function. Comparing FDA results with other optimization algorithms shows better performance of FDA in solving challenging problems. Kivi et al. proposed a nature-inspired transcendental algorithm called Sheep Flock Optimization Algorithm (SFOA) that mimics shepherds and sheep behaviors in pasture [59]. Experimental results show that SFOA is superior to the most advanced meta-heuristic algorithms such as SSA, SBO, GWO as well as conventional meta-heuristic methods such as GA, PSO, ICA.

Regardless of differences between meta-heuristics, two major characteristics are common for all including exploration (diversification) and exploitation (intensification) [60]. By heuristics, we mean global search. In order to support this step, an algorithm must have random operators so that it can search that space randomly and globally. Exploration or exploitation means obtaining the better responses. These two factors have conflicts and are like some obstacles for each other. Therefore, by considering the logical and suitable balance between them, an appropriate optimizer should be designed.

According to No Free Lunch (NFL), it is logically proved that no metaheuristic can respond to all correct optimization problems. In other words, the especial metaheuristic method can demonstrate promising results to solve some problems. But this algorithm may show the weaker performance and efficiency for some other problems. Hence the researchers try to create the better optimization techniques each year.

This paper proposes a novel evolutionary algorithm without recombination operator. It uses novel clonal and conservative selection operators. In the proposed algorithm, the initial population is generated through the chaos theory for better exploration. A temporary target is determined in each intelligent iteration in ICO. Each parent is cloned on the basis of its fitness, and each offspring is located near the parent or in the route of temporary target. It is different in various iterations in terms of how many offsprings are created by the first method or by the second method. They are controlled by algorithm parameters.

The offsprings are classified to two classes according to generate method. In ICO, a conservative selection operator is proposed. By using this operator, the novel population is selected from two classes of offsprings and existing population by maintaining the population diversity. The proposed operators used in ICO algorithms can be effective in increasing the balance between exploitation and exploration.

The proposed algorithm was evaluated on 39 well-known benchmark functions of unimodal, multimodal, fixed-dimensional multimodal, composite and CEC 2019 to test exploration, exploitation, local optima avoidance, scalability and convergence properties. Furthermore, ICO is used to solve three engineering application problems. The obtained results of the proposed algorithm are compared with the results of sixteen state-of-art metaheuristic algorithms. According to the comparison results, the proposed algorithm outperformed the conventional and state-of-the-art methods.

The rest of the paper is structured as follows. Section 2 describes the Intelligent Clonal Optimizer developed in this study. Then Section 3 discusses experimental results and its comparison with other popular techniques. ICO, in Section 4, solves three engineering application problems. Finally, conclusion and future work are shortened in Section 5.

2. Intelligent clonal optimization (ICO) algorithm

This paper proposes an intelligent evolutionary algorithm with a novel clonal and conservative selection operator, and the solutions are initialized through the chaos theory. Cloning the solution is performed on the basis of its fitness. The solutions are cloned locally and in trajectory of temporary target. For the better population convergence to the found best solution, an interval was selected to measure proximity to a target; thus, the search process is continued in designated interval. When all solutions are too close to the temporary target, new random solutions are added to the population for the better population convergence to the found best solution. The different steps of the proposed algorithm are the followings:

(1) Initializing parameters
(2) Generating the initial population through the chaos theory
(3) Determining the fitness of solutions
(4) Applying the proposed clonal operator
   a. Determining the number of clones for each parent
   b. Determining the temporary target
   c. Cloning the parents intelligently both locally and in the direction of temporary target
(5) Deleting similar offsprings from each list of new offsprings (XL, XB) and the parents
(6) Selecting elite offsprings from the parents and each list of new offsprings for transition to the next generation
(7) Repeating Steps 4–7 until the termination condition
(8) Selecting the best member of the last generation as the algorithm solution and ending the algorithm

2.1. Initializing the initial population through the chaos theory

Evolutionary algorithms (EAs) can use chaotic characteristics-randomness, ergodicity and regularity to avoid falling into local minima in the search process [61,62]. In addition, it is effective in terms of speeding the convergence [63].

As one of the simplest chaotic maps, logistic map is considered by the researchers in recent decades [64]. In the proposed algorithm, the initial population is generated through the chaos theory and logistic mapping. In a chaotic system with a logistic mapping function, if any of the final points is selected to start a path, the path will finally converge on the starting point. The following equation shows the logistic mapping [65].

$$X_{n+1} = \mu \times X_n \times (1 - X_n), \qquad n = 0, 1, \ldots \tag{1}$$

In this equation, μ is the control parameter. When $\mu = 4$ and $X_0 \in [0, 1]$, the logistic function is totally chaotic and includes the entire [0, 1] interval. Chaotic variables can be converted into optimization variables through the following linear mapping:

$$X_n = a + (b - a) \times X_n \tag{2}$$

In this mapping, a and b refer to the lower and upper boundaries of optimization variables respectively.

First, n-dimensional vector ($X_1 = X_{11}, X_{12}, \ldots, X_{1n}$) is created randomly within the [0, 1] interval through a normal distribution to generate the initial population. In this vector, n shows the number of decision variables. Then, n chaotic vectors ($X_1, X_2, \ldots, X_n$) are created through the logistic equation (1) by considering $\mu = 4$. Each component of X_i is then mapped onto the interval corresponding to that variable through the mapping equation (2).

2.2. Novel clonal operator

In the proposed method, all solutions are considered as the parent in existing population, and then they are cloned. Hence, the number of clones is firstly determined for each parent, and then each parent is cloned on the basis of determined number, and the offspring is generated.

The fitness values of parents are used to determine the number of offsprings generated from each parent. The number of cloned parents and the method of cloning are discussed in the following subsections.

2.2.1. Number of cloned parents

The number cloned parents can be calculated through the following equation based on their fitness.

$$S_i = S_{min} + (S_{max} - S_{min}) \times NF_i \tag{3}$$

$$NF_i = \frac{f_i - f_{worst}}{f_{best} - f_{worst}} \tag{4}$$

Where, S_i is the number of offsprings generated from $parent_i$. S_{min} and S_{max} show the minimum and maximum cloning rates respectively. Moreover, NF_i indicates the normalized fitness of each solution with f_i referring to the solution fitness. Furthermore, f_{worst} and f_{best} show the worst and best fitness values in the populations respectively. In a minimization problem, the best fitness value is the lowest fitness value, whereas the worst fitness value is the highest fitness value. However, a maximization problem follows an inverse procedure.

2.2.2. Parent cloning

In this step, the parents are cloned in two ways. Few offsprings are generated next to their parents, whereas others are generated in the trajectory of the temporary target, and they are placed in XL and XB list.

The distance of each offspring with its parents are determined through Eq. (5). The initial values of ΔX are considered zero for all solutions.

$$\Delta X_i = r \times \Delta X_{i-1} + A_i \tag{5}$$

Where r is a random number in [0 1], and the value of A_i is obtained from localization of elite solutions.

In order to compute A_{i}, the better solution is selected from population on the basis of n_{iter} number, and they are placed in ES list according to Eq. (6). The value of n_{iter} with normal distribution is obtained by Eqs. (7)–(8). According to this equation, n_{iter} is reduced from a previously defined initial value to a final value, 2 percent of solutions, in every step.

$$ES = \{x_k | x_k \in X, \quad X = \underset{x}{\operatorname{argsort}} f(x), \ 1 \le k \le n_{iter}\} \tag{6}$$

$$y = round\left(N \times \left[98 \times \left(1 - {}^{\mathrm{iter}}/_{\mathrm{k}}\right) + 2\right]/100\right) \tag{7}$$

$$n_{iter} = \begin{cases} y & y \ge 1 \\ 1 & otherwise \end{cases} \tag{8}$$

The value of k depends on the number of iterations. In the proposed method, the number of cloning each parent is not fixed, and it depends on the fitness of that parent and current iteration. Also it is required to reach MaxFEs to end the algorithm so an exact value cannot be stated for the number of iterations, while maximum and minimum number of iterations can be estimated, and k value can be specified according to them. The value of k can be computed by the following equation.

$$k = 0.25 \times \left(\frac{MaxFEs}{Smax \times N} + \frac{MaxFEs}{1 \times N}\right) \tag{9}$$

$$= 0.25 \times \frac{MaxFEs(1 + Smax)}{Smax \times N} \tag{10}$$

Where MaxFEs is maximum number of fitness function evaluation. N refers to the number of initial population, and Smax is equal to maximum number of clones. A_i is obtained from the following equation:

$$A_i = 20 \times \alpha_{iter} \times E_i \tag{11}$$

$$\alpha_{iter} = 10 \times (log(M)) \times Z \tag{12}$$

$$Z = exp(-\beta \times iter/k) \tag{13}$$

$$M = (Range\,(2) - Rang(1))/2 \tag{14}$$

Where M represents the mean variation interval of the variable. Range is the function boundary of search space. Therefore, Range (1) and Range (2) are respectively minimum and maximum values for x variable. For example, with regard to F1 function, according to Table 2, line 1 and row 3, Range (1) and Range (2) are respectively −100 and 100.

An interval is considered for better exploration in Z, and to prevent an offspring approaching too close to a temporary target, and to control the search interval. Whenever Z is exceeded by γ, β is altered through Eq. (15) to change Z to $10 \times \gamma$ and to continue the search in this interval. β_0 and γ are constants with values equal to 100, 1e−19.

$$\beta = \begin{cases} -\log(10 \times \gamma) \times \dfrac{k}{iter}, & Z \le \gamma \text{ and } iter > 1 \\ \beta_0, & iter = 1 \end{cases} \tag{15}$$

The value of E_i is obtained from the following equation.

$$E_i = \frac{TT_i - X_i}{R + \in} \tag{16}$$

$$TT_{i,d} = \frac{\sum_{j \in \mathrm{ES}} \left(r_i \times NF_j\right).X_{j,d}}{n_{iter}} \tag{17}$$

where TT is a temporary target, R shows the Euclidean distance between X_{i} and TT_i with a norm of 2, $\in$ is a positive small constant, whereas $\mathrm{NF_j}$ shows the normalized fitness of $\mathrm{Solution_j}$ (determined through Eq. (4)), and ES represents the set of selected elite solutions. Furthermore, d shows the problem dimensions, whereas r_{i} is a random constant in [0 1]. This random number was considered as a random constant to calculate all dimensions of TT for solution; however, it is changed for another solution.

The probability of generating a offspring around its parent is σ_{iter} so the probability of generating a offspring in trajectory of the temporary target is $1 - \sigma_{iter}$. The value of σ_{iter} with a normal distribution can be obtained by Eq. (18). This step aims at generating a larger number of offsprings with a normal distribution, transferring a larger percentage of them to the next generations and reducing this number later.

$$\sigma_{iter} = \left[{}^{(k-iter)^{Ex}}/_{(k-1)^{Ex}}\right] \times \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{18}$$

where $\sigma_{initial}$ shows the initial value of standard deviation, and σ_{final} indicates the final value of standard deviation. Moreover, σ_{iter} is the standard deviation in the current iteration, Ex shows the nonlinear modulation index, and k is determined through Eq. (9). After σ_{iter} is determined, two groups of offsprings are generated in this step by Eq. (19). A few of them are generated next to the parent in a list named XL, whereas a few of them are generated in the trajectory of temporary target in another list named XB.

$$\begin{cases} XB_{ib} = X_i + \Delta X_i, \quad XB\Delta X_{ib} = \Delta X_i, \\ \qquad 1 \le ib \le NB \qquad if \quad r \ge \sigma_{iter} \\ XL_{il} = X_i + \alpha_{iter} \times rn, \quad XL\Delta X_{il} = \Delta X_i, \\ \qquad 1 \le il \le NL \qquad if \quad r < \sigma_{iter} \end{cases} \tag{19}$$

It should be mentioned that r is a random number in [0 1], whereas X_i is parent i, and ΔX_i indicates the distance between the parent and offspring is calculated by Eq. (5).

The value of ΔX is saved in $XB\Delta X$ so that it can be selected and used to calculate next iterations. The offspring is also saved in a list named XB.

The offsprings generated next to the parent with a probability of σ_{iter} will be determined through Eq. (18) where the distance between the parent–child decreases in each step.

In this equation, rn represents random real numbers with the mean of 0 and standard deviation of 1, whereas α_{iter} refers to every iteration of values calculated through Eqs. (12)–(15). The value of ΔX for each parent is saved in $XL\Delta X$ so that it can be selected and used to calculate next iterations.

2.3. Conservative selection operator

The offsprings are generated by two different methods in Section 2.2, and they are placed in two separate lists of XB and XL. In this step, the members at that are similar each other are deleted from each list of XL, XB and parents. Then NL, NB and NE elite solutions are selected from each list of XL, XB and the parents.

$$NL = \begin{cases} N_{XL}, & N_{XL} \le \lceil(\sigma_{iter}) \times N\rceil \\ \lceil(\sigma_{iter}) \times N\rceil, & otherwise \end{cases} \tag{20}$$

$$u = N - NL \tag{21}$$

Table 2
Unimodal benchmark test functions.

Function	D	Range	F min
$F_1(X) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(X) = \sum_{i=1}^{n} \lvert X_i \rvert + \prod_{i=1}^{n} \lvert X_i \rvert$	30	[−10,10]	0
$F_3(X) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	[−100,100]	0
$F_4(X) = \max_i \{ \lvert x_i \rvert .1 \le i \le n \}$	30	[−30,30]	0
$F_5(X) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30	[−100,100]	0
$F_6(X) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$F_7(X) = \sum_{i=1}^{n} i x_i^4 + random[0.1)$	30	[−1.28,1.28]	0

$$NB = \begin{cases} N_{XB}, & N_{XB} \le \lceil u \times 0.9 \rceil \\ \lceil u \times 0.9 \rceil, & otherwise \end{cases} \tag{22}$$

$$NE = \begin{cases} N_X, & N_X \le u - NB \\ u - NB, & otherwise \end{cases} \tag{23}$$

In these equations, N_{XL} and N_{XB} refer to the number of members in XB and XL lists. N_X refers to the number of parents. According to Eq. (20), the value of NL that is the number of offsprings selected from XL list is specified. *N* is defined as the number of the initial population. Accordingly, a larger number of offsprings are selected from XL list in the first iterations; however, there are gradually fewer offsprings in next iterations. The number of remaining members are determined so that they can be located in new generation according to Eq. (21). Then 90 percent of them are computed as NB to be selected from XB according to Eq. (22), and others are computed as NE to be selected from parents according to Eq. (23) (in order to maintain elitism).

One unit is subtracted from NB to select at least one elited member from the parents when NE equals to 0 based on calculations if $NB > 1$; otherwise, one unit is subtracted from NL, and NE changes to 1.

Moreover, when the number of selected member is smaller than N, random members are added to the new population to maintain population diversity. After the number of selected members is determined in each list, the new population is selected. In fact, the new population includes the selected elited solutions in each list. The previous ΔX values of each member are also transferred to the next step for calculations in the next iterations.

2.4. Algorithm termination condition

In metaheuristic algorithms, a different termination can be defined. For instance, a termination condition can be a specific number of iterations, constraints on the number of function evaluations, a specific rate of precision, a specific time, no sign of change in solutions after a specific number of iterations, or a combination of these cases. In the proposed method, the termination condition was defined as imposing constraints on the number of function evaluations (MaxFEs) and reaching a specific rate of precision.

Algorithm 1 shows the pseudocode of the algorithm, while Fig. 1 presents a flowchart. The source codes of this algorithm can be found in https://github.com/MSNFV/IntelligentClonalOptimizer.

2.5. Time complexity

The time complexity of the proposed method is as follows in different steps.

Initialization of population: $O(N \times D \times I)$
Computing temporary target: $O(N \times D \times I)$
Computing the vector A: $O(MaxFEs \times D)$
Computing fitness function for each solution: $O(MaxFEs \times D)$
Selection step: $O(MaxFEs \times N)$

Where *N* is the size of initial population. D is dimension of the problem, and MaxFEs is maximum number of function evaluations. *I* stands for the number of iterations in the proposed method. Since S number is produced offspring for each parent in $[S_{min} S_{max}]$ in each iteration, it can be concluded that the value of $I \times N$ is less than MaxFEs. Therefore, overall time complexity of the proposed algorithm is equal to $O(MaxFEs \times (D + N))$.

2.6. ICO vs. comparative algorithms

The metaheuristic algorithms are different in various steps on the basis of their used operators. In the proposed algorithm, three operators including initial initialization, cloning and selection are used. In this section, differences and similarities of these operators with similar operators in other metaheuristic methods are presented.

- Initialization Operator
  The initialization is randomly performed in most evolutionary algorithms like genetic algorithm [35,38]. Chaos algorithm is used in some algorithms for initialization [65]. In this paper, we consider chaos theory for initialization, but ICO operators are totally different from the operators of other algorithms.
- Clonal Operator
  In most of evolutionary algorithms, like genetic algorithm, crossover operator is considered, and clonal operator is rarely used. The clonal operator is used in some algorithms like IWO [38] and AIS [66]. The difference between these methods and the ICO is how they are cloned. Clone of these two methods is based on normal distribution, while clone of the ICO is done in two different ways with different formulas (5–19).
  Some are cloned near the parent, and some are cloned by temporary target and the distance between the parent and temporary target. On the other hand, proposed method used

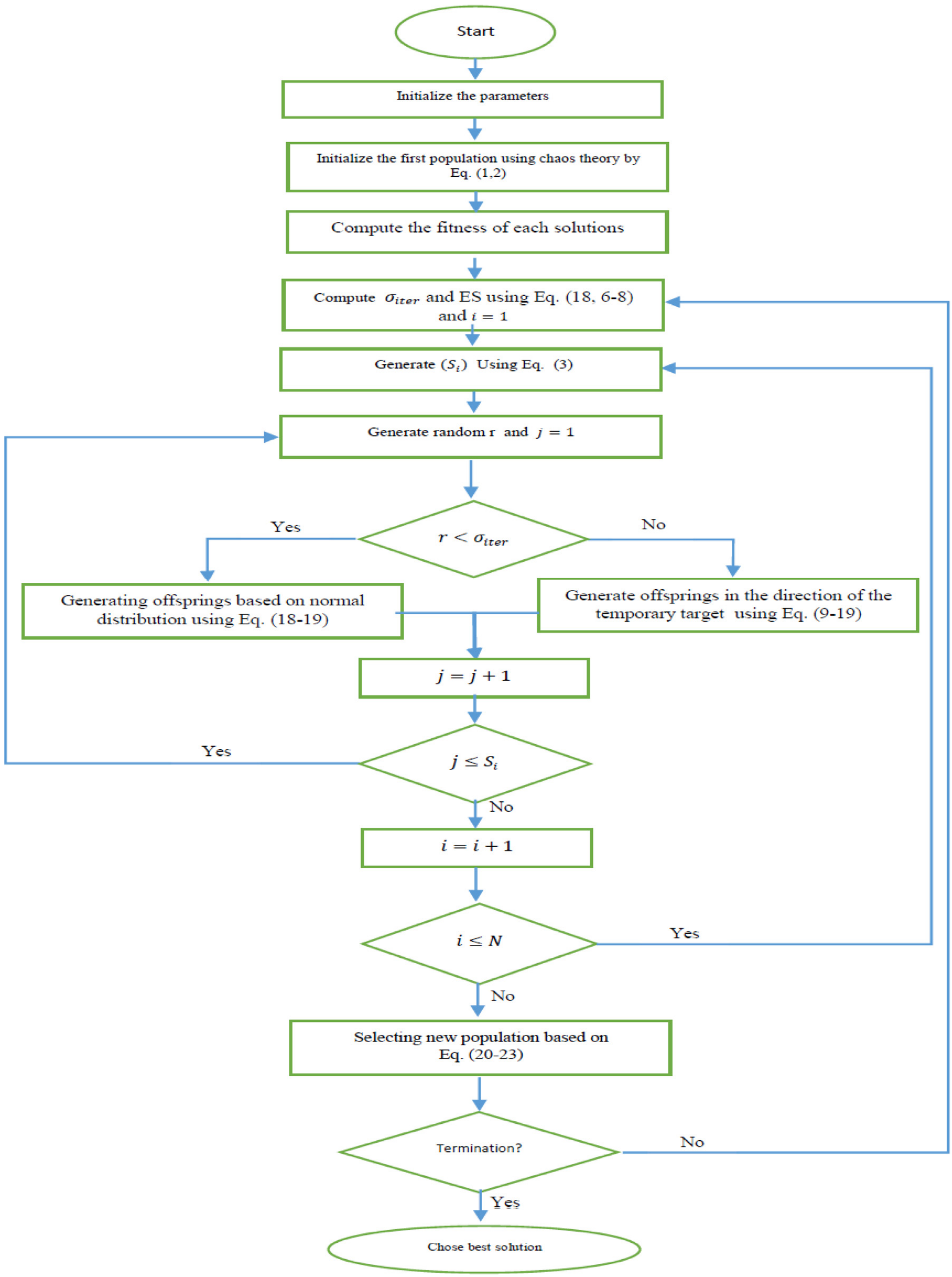

Fig. 1. The flowchart of ICO.

a new idea in order to prevent being trapped in local optimality; accordingly, an interval has been defined in order to adjust proximity degree of offspring to temporary target, while this approach did not exist in similar methods.

- Selection Operator
In most of evolutionary algorithms, like genetic algorithm [38], the new population is the elited solutions of previous population in selection step. In other algorithms, like evolutionary strategies ((μ + λ)-ES), the new population is the elited solutions of parents and offsprings population [37]. In the proposed method, the new population is separately selected from three classes including the most elited solutions of the parents and each list of new offsprings (XL and XB). by maintaining the population diversity.

Algorithm 1 ICO

Procedure Inteligent Clonal Optimizer
Define input parameters (N, MaxFEs)
$\sigma_{initial} = 0.5,\ \sigma_{final} = 0.1, S_{min} = 0, S_{max} = 40$
$\alpha_0 = 100, \gamma = 1e-19$, μ = 4, Ex=2
Create the initial population $[X]_{N\times n}$ using Eq. (1-2)
Evaluate fitness of each chromosome (F_1,…, F_N) by objective function
Iter=1; FEs=N;
Compute the k using Eq. (10)
Compute M using Eq. (14)
while the stopping criterion is not satisfied **do**
  Compute the normalized fitness vectors (NF_1,…, NF_N) using Eq. (4)
  Compute the S vectors (S_1,…, S_N) using Eq. (3)
  Compute the n_{iter} using Eq. (7-8)
  Compute the ES vectors (ES_1,…, ES_k) using Eq. (6)
  Compute σ_{iter} using Eq. (18)
  Update β using Eq. (15)
  Update D using Eq. (13)
  Compute α_{iter} using Eq. (12)
  Compute the E vector using Eq. (16-17)
  Compute the A vectors using Eq. (11)
  il=1; ib=1;
  for i = 1 to N **do**
    for j=1 to S_i **do**
      r = rand(0, 1)
      if $r < \sigma_{iter}$ **do**
        $XL_{il} = X_i + \alpha_{iter} \times rn$,
        $FL_{ib} = ObjectiveFunction(XL_{ib})$
        $XL\Delta X_{il} = \Delta X_i$
        il=il+1
      else
        $\Delta X_i = r \times \Delta X_{i-1} + A_i$
        $XB_{ib} = X_i + \Delta X_i$,
        $FB_{ib} = ObjectiveFunction(XB_{ib})$
        $XB\Delta X_{ib} = \Delta X_i$
        ib=ib+1
      end if
      FEs=FEs+1;
    end for
  end for
  Updatte X, F and ΔX vectors by the selection mechanism based on Eq. (20-23)
  iter=iter+1
end while
Return the best optimal solution
end procedure

3. Experimental studies

For efficiency analysis, the proposed algorithm was evaluated on 39 different optimization benchmark functions and three real-world engineering problems. The benchmark functions are listed in Tables 2–6, where D indicates dimension of the function, Range denotes the range of variation of optimization variables, and f_{min} is the optimum. These benchmark functions can be organized into five categories including unimodal, multimodal, fixed-dimension multimodal, composite and CEC2019. In this section, the comparison methods included three categories of optimization methods:

- The EO [1], SASS [2], GWO [8], WOA [13],SSA [20], MFO [10], GSA [42], SHO [19], AOA [55] and HGSO [67] methods are proposed as recently developed algorithms.

Table 3
Multimodal benchmark test functions.

Function	D	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	$-418.9829 \times d$
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[−5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1 + 10\sin 2(\pi y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[−50,50]	0
$y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i-a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i-a)^m x_i < -a \end{cases}$			
$F_{13}(x) = 0.1\{\sin 2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1 + \sin 2(3\pi x_i + 1)\right] + (x_n-1)^2\left[1 + \sin 2(2\pi x_n)\right]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0

Table 4
Fixed-dimension multimodal benchmark test functions.

Function	D	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum(x_i - a_{ij})^6}\right)^{-1}$	2	[−65,65]	0.998
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[−5,5]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 + 36x_1x_2 + 27x_2^2)\right]$	2	[−2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{i=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{i=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.5363

- The DE [39] and PSO [43] methods were selected as the most well-known evolutionary and swarm intelligence algorithms.
- The GA_MPC [68], LSHADE [69], LSHADE–cnEpSin [70], and LSHADE–SPACMA [71] methods were selected as the best performer of the IEEE CEC competitions. In fact, GA–MPC won CEC2011, whereas LSHADE won CEC2014. Finally, LSHADE–cnEpSin, and LSHADE–SPACMA won CEC2017.

The evaluation process was performed in MATLAB R2019(b) software platform under the Windows 10 system with an Intel (R) Core (TM) i7 CPU Q720 @ 1.6 GHz 1.6 GHz and 4 GB RAM. To draw a fair comparison between the proposed algorithm and other methods, allowed maximum number of function evaluations (MaxFEs) was set D × 10,000 for all algorithms (D is the problem dimensions). The number of initial population was considered as 30 for all algorithms. In addition, the control parameters of all algorithms are set as recommend setting in their original papers. Table 7 shows the setting of parameters for each algorithm. The numbers of independent algorithmic runs are equal to 30 utilized to generate the statistical results.

3.1. Parameter sensitivity analysis

This section analyzes the sensitivity of control parameters of ICO, which are Smax, $\sigma_{initial}$, and σ_{final}. Some functions are picked

Table 5
Composite benchmark test functions.

Function	D	Range	f_{min}
F_{24}(CF1) $f_1, f_2, \ldots, f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	10	[−5,5]	0
F_{25}(CF2) $f_1, f_2, f_3, \ldots, f_{10}$ = Criewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	10	[−5,5]	0
F_{26}(CF3) $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1, 1, 1 \ldots, 1]$	10	[−5,5]	0
F_{27}(CF4) f_1, f_2 = Ackley's Function, f_3, f_4 = Rastrigin's Function, f_5, f_6 = Weierstrass Function, f_7, f_8 = Griewank's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ $= [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	[−5,5]	0
F_{28}(CF5) f_1, f_2 = Rastrigin's Function, f_3, f_4 = Weierstrass Function, f_5, f_6 = Griewank's Function, f_7, f_8 = Ackley's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ $= [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	[−5,5]	0
F_{29}(CF6) f_1, f_2 = Rastrigin's Function, f_3, f_4 = Weierstrass Function, f_5, f_6 = Griewank's Function, f_7, f_8 = Ackley's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	10	[−5,5]	0

Table 6
CEC2019 test functions.

Function	D	Range	f_{min}
F30 (CEC01_2019) : Storn's Chebyshev polynomial fitting problem	9	[−8192,8192]	1
F31 (CEC02_2019) : Inverse Hilbertmatrix problem	16	[−16384,16384]	1
F32 (CEC03_2019) : Lennard-Jonesminimum energy cluster	18	[−4,4]	1
F33 (CEC04_2019) : Rastrigin'sfunction (F9(x))	10	[−100,100]	1
F34 (CEC05_2019) : Griewank'sfunction (F11(x))	10	[−100,100]	1
F35 (CEC06_2019) : WeierstrassFunction	10	[−100,100]	1
F36 (CEC07_2019) : ModifiedSchwefel's function	10	[−100,100]	1
F37 (CEC08_2019) : ExpandedSchaffer's F6 function	10	[−100,100]	1
F38 (CEC09_2019) : Happy CatFunction	10	[−100,100]	1
F39 (CEC10_2019) : Ackley function(F10(x))	10	[−100,100]	1

from each category of functions (i.e., F1, F8, and F14). The number of initial populations considered as 30 and MaxFEs is equal to $D \times$ 10,000 in performed evaluations, and the mean of 30 independent runs are demonstrated in diagrams.

In Fig. 2, diagram of ICO algorithm convergence is shown by considering different values of (2, 5, 10, 20, 30, 40) for Smax parameters. The values of other parameters are set in these evaluations according to Table 7.

The number of parent cloning (S) decreases automatically in each iteration so that convergence is facilitated. The initial value is Smax, and its final value is considered as zero equal to Smin. The number of clones increases by increasing Smax. The value of Smax can affect convergence speed. In the multi-modals and complex problems, convergence speed increases when the value of Smax increases since more offspring are produced when the value of Smax increases in each iteration and ICO can escape falling into local optimum by precise search. In unimodal problems, by reducing the value of Smax, ICO has the higher convergence speed because the computation of fitness function is decreased in each iteration by reducing Smax. Since the maximum number of computing fitness function is limited so the number of iterations increases. ICO can be close to the objective when the number of iterations increases.

Totally, in simpler problems with single purpose, it is better to have smaller Smax value so that it can quickly reach the purpose. In more complicated problems with several purposes, it is better to assign higher Smax values so that it can have the better search and can escape the local optimum. According to Fig. 2, the value of Smax is a suitable selection in the range of 2–10 for unimodal functions. The value of Smax in the range of 20–40 is a suitable selection for other functions. The value of Smax is equal to 2 for unimodal functions, and it is considered as 40 for other functions.

σ_{iter} is the standard deviation in the current iteration in the range of $[\sigma_{initial} \quad \sigma_{final}]$, Where $\sigma_{initial}$ shows the initial value of standard deviation, and σ_{final} indicates the final value of standard deviation. According to the value of σ_{iter}, the offspring is produced around the parent. This value is decreased automatically in each iteration. By reducing According to the value of σ_{iter} in each iteration, the probability of producing the offspring increases in temporary target direction. The value of σ_{iter} is equal to σ_{initial} in the first iteration, and it is equal to σ_{final} in the last iteration. When the value of σ_{initial} is high, the offsprings are produce

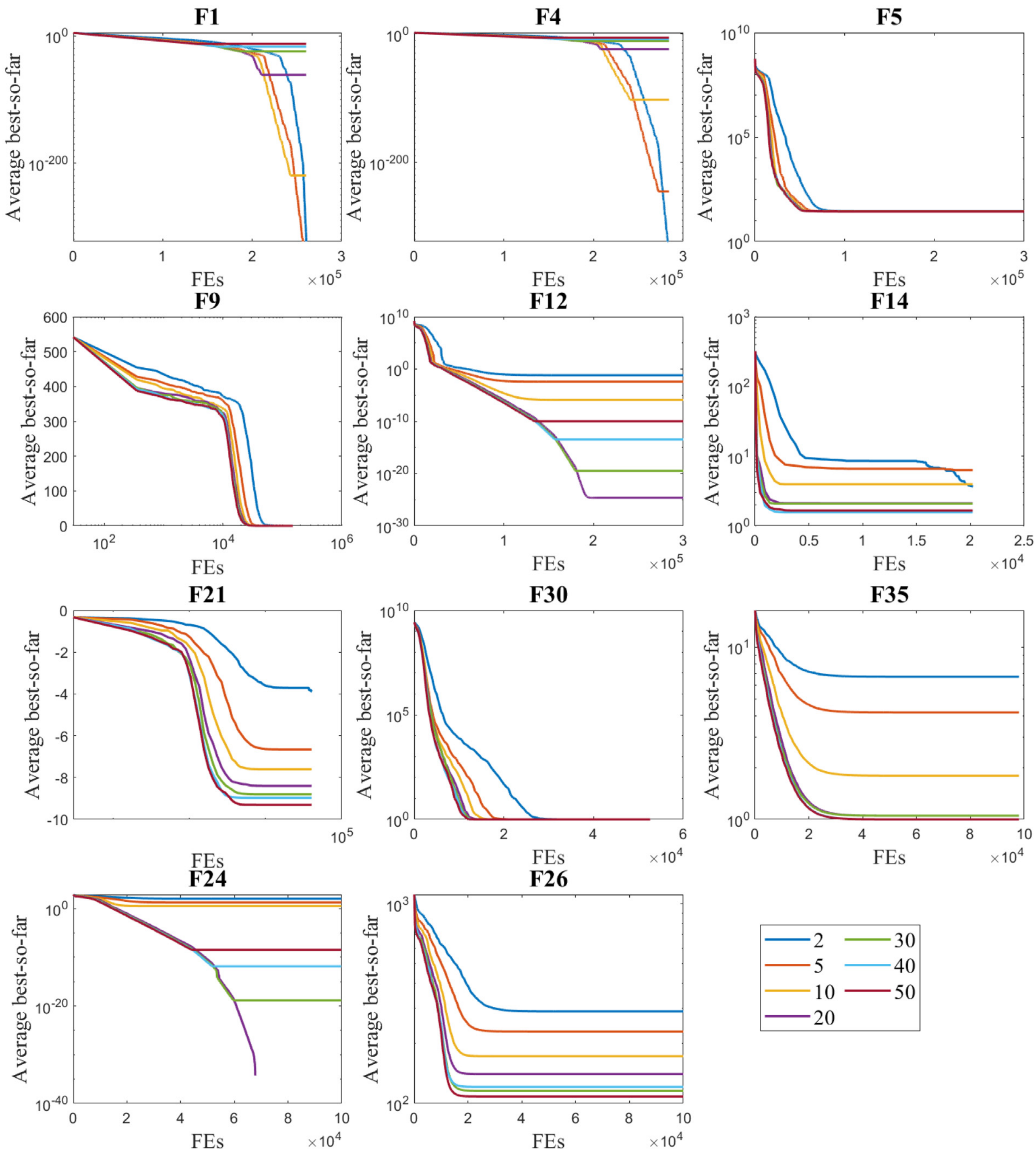

Fig. 2. Sensitivity analysis of ICO's control parameter Smax with different functions.

around the parent with the higher probability, and less offsprings are produce in temporary target direction. When the value σ_{final} is less, fewer offsprings are produced around the parent so most of offsprings are produced in temporary target direction. The values should be considered for these two parameters so that children are produced in two ways in different iterations, and the algorithm can be close desirable result by suitable searching. The values of two parameters, $\sigma_{initial}$ and σ_{final}, are dependent to each other. Therefore, we evaluated ICO for these two parameters by considering different values. The value of σ_{final} should be smaller than $\sigma_{initial}$. The evaluations are performed by considering the values of {0.3, 0.5 0.7, 0.9} for $\sigma_{initial}$ and the values of {0.1, 0.3, 0.5, 0.7} for σ_{final}. By selecting a value for $\sigma_{initial}$, the smaller values are investigated for σ_{final}. In these evaluations, the values of other parameters are set according to Table 7. The results of evaluation are demonstrated in Fig. 3. According to Fig. 3, the values between 0.3–0.7 and 0.1–0.5 are respectively the suitable selections for, $\sigma_{initial}$ and σ_{final}. In our evaluations, the values of $\sigma_{initial}$ and σ_{final} are respectively considered as 0.5 and 0.3.

3.2. Performance comparison

For performance comparison analysis, the proposed algorithm was evaluated on 39 different optimization benchmark functions. The benchmark functions included unimodal, multimodal, fixed-dimension multimodal, composite and CEC2019. The performance evaluation indexes of experiments are considered as

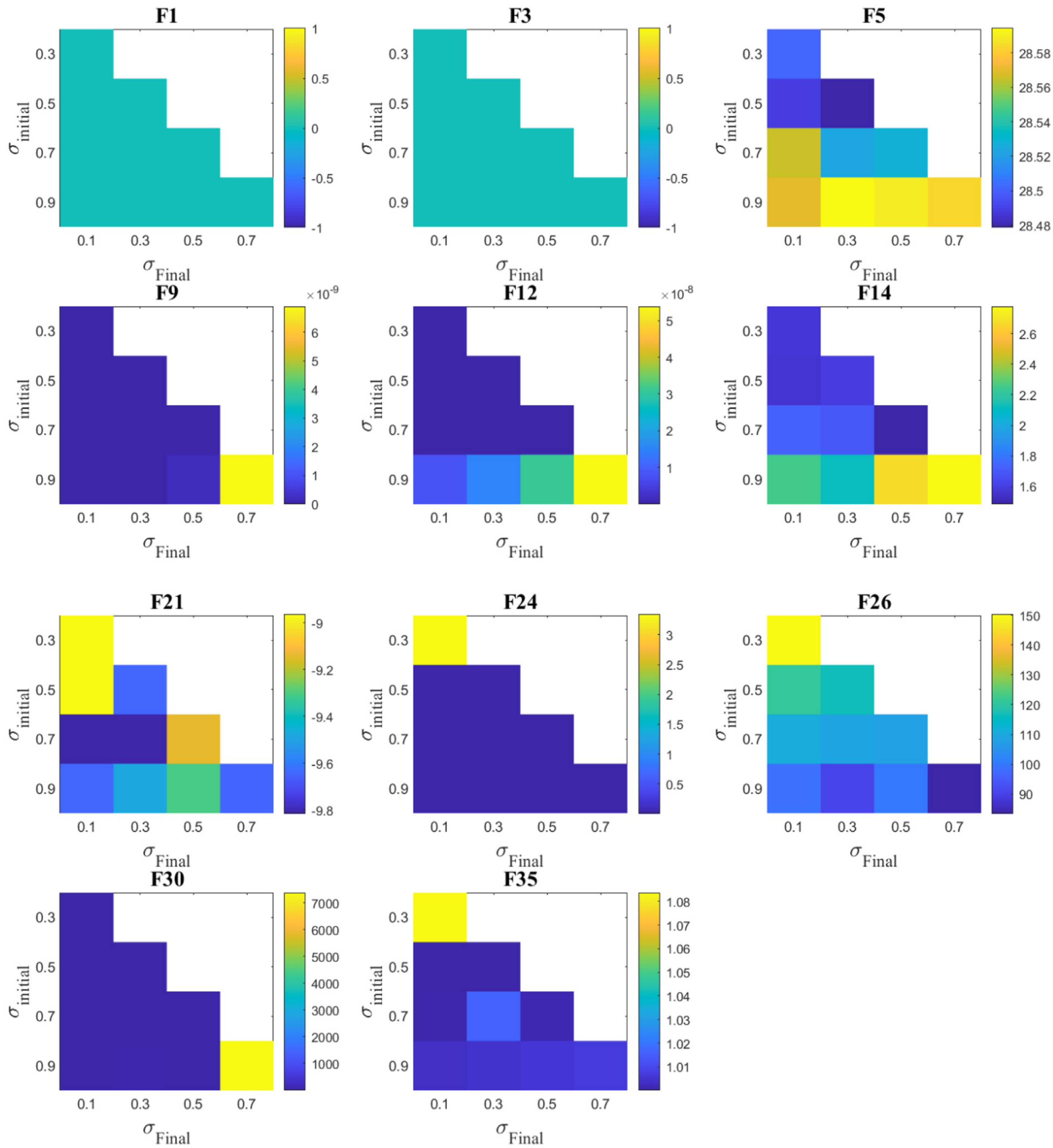

Fig. 3. Sensitivity analysis of ICO's control parameters $\sigma_{initial}$ and σ_{final} with different functions.

follows: best (Best), mean (Mean) and standard deviation (Std) over 30 runs.

3.2.1. Evaluation of 30-dimensional unimodal test functions F_1–F_7

Functions F1–F7 belong to the unimodal class of mathematical functions having one single solution to evaluate exploitation of optimization methods. Table 2 shows these functions along with their dimensions. Table 8 presents the results of evaluating the proposed method through unimodal functions as opposed to the existing methods.

According to the obtained results, ICO and HGSO could reach the global optimal point (0) in all 30 runs with superior performance in five functions including F1, F2, F3, F4 and F6. AOA algorithm could reach the global optimal point (0) in four functions including F1, F2, F3 and F6. Also, after AOA, ICO could reach the best results in F7 in comparison to other methods. EO algorithm could reach the global optimal point (0) in two functions including F1 and F6, while GWO and WOA could reach that point in three functions involving F1, F2 and F6, and LASHDE-SPASMA could reach that point in F1 function. The algorithms of PSO, DE and GSA could only reach this point in function F6. According to Wilcoxon Signed-Rank test in Table 13, it can be said that ICO presents the equal or the better results than other methods in F1, F2, F3, F4 and F6. In addition, it obtained the better results in F7 than other methods except AOA.

Furthermore, the proposed method could obtain the best possible results in most of the functions in this category. According to Fig. 5, it converged faster than most of the methods; as a result, it can be claimed that the proposed method has high exploitation capacity.

Table 7
Parameter settings for algorithms.

Algorithm	Parameters
ICO	Ex, μ, β_0, γ are constants with values equal 2, 4, 100, 1e−19, $\sigma_{initial} = 0.5, \sigma_{final} = 0.1$ $S_{\min} = 0$, $S_{\max} = \begin{cases} 2, F \in unimodal\ Function(F1 - F13) \\ 40, otherwise(F14 - F39) \end{cases}$
EO	$a_1 = 2, a_2 = 1, GP = 0.5$
GWO	$parameter(a) =$ [20]
WOA	$a_1 =$ [20]; $a_2 = [-2 - 1]; b = 1$
MFO	$Convergence\ constant(r) = [-1 - 2]$ $Spiral\ factor\ (b) = 1$
PSO	$Cognitive = 2, social = 2$ $Inertial\ weight = [0.90.4]$
DE	$Scaling\ factor = 0.5, Crossover\ probability = 0.5$
GSA	$Rnorm = 2, Rpower = 1, alpha = 20, andG0 = 100$
SSA	$Leader\ position\ update\ probability = 0.5$
SHO	$predators = [0.7, 0.9]$
GA-MPC	$\beta = N(0.7, 0.1), cr = 1, p = 0.1, tornoment\ pool\ size = [2, 3]$
LSHADE	$Pbest = 0.11, Arcrate = 1.4, Memory\_size(H) = 5$
LSHADE-cnEpSin	$Nmin = 4, Nmax = D \times 10e4, D = Dimension\ size \mu_f = 0.5, \mu_{cr} = 0.5, H = 5, ps = 0.5, pc = 0.4$
LSHADE-SPACMA	$Learning\ rate\ (c) = 0.8, threshold = 10e{-8}$ $H = 5, Pbest = 0.11, Arcrate = 1.4$ $Probability\ variable(Fcp) = 0.5$
SASS	$Nmin = 4, Nmax = D \times 10e4, D = Dimension\ size$ $PbestRate = 0.11, rd = 0.5, c = 0.7, Arcrate = 1.4, Ms = 5$

3.2.2. Evaluation of 30-dimensional multimodal test functions F_8–F_{13}

The functions F8–F13 belong to the multimodal category. They are employed to evaluate the exploration of optimization methods. These functions have more than one optimal solution. Table 3 indicates the details on this category of functions. Table 9 includes the results of evaluating the proposed algorithm as opposed to other methods in this category of functions.

According to the obtained results, ICO algorithm along with EO and HGSO algorithms could reach the global optimal point by the superior performance in functions F9 and F11 in all 30 runs. GWO and AOA could reach this point in F9, while WOA could reach global optimal point in F11. DE algorithm, in comparison to PSO and Eo, reached the better result in F12 with less differences, while it reached the better result in F13 with less differences in comparison to other methods. HGSO and AOA algorithms had the better results in the functions F8 and F10 respectively in comparison to other algorithms. The results reveal that ICO has also good exploration capability.

3.2.3. Evaluation of fixed-dimension multimodal test functions F_{14}–F_{23}

Functions F14–F23 are categorized as the fixed-dimension multimodal category. Like the previous category, these functions are used to evaluate the exploration evaluation of optimization methods; however, their dimensions are low and fixed.

Table 4 indicates the fixed-dimension multimodal test functions. According to the results presented on Table 10, the proposed method could yield the best possible results in all functions of this category, except for F15.

Regarding the mean result, the proposed method could reach the better result along with most algorithms in functions F17–F19. Totally it can be said that DE algorithm has the better performance and higher exploration capacity in these functions, and it has the better results in most functions than other methods.

3.2.4. Evaluation of composite test functions F_{24}–F_{29}

The composite functions (F24–F29) imitate an actual search domain by having a large number of local optimums and complexity. These functions are very challenging because it is possible to escape from the local optimum only by establishing an appropriate equilibrium between exploration and exploitation. The technical report CEC 2005 presented more details on the integration performance. Table 5 presents more details on this category of functions, and Table 11 presents the results of evaluating the proposed method in the functions of this category. By considering the mean of obtained results, the proposed method performed well in comparison with other methods of this category because it could obtain the better result in 4 out of 6 functions (F25, F27, F28, and F29) than other methods.

With regard to the results' mean, ICO obtained the second rank in function F24 after GSA. In addition, SSA reached the best result in function F26, and GSA and ICO respectively obtained the second and third rank in comparison to other algorithms.

The evaluation results indicate the high efficiency of ICO in terms of solving high-complexity problems with many local optima, and it can balance exploration and exploitation phases.

3.2.5. Evaluation of CEC 2019 test functions F_{30}–F_{39}

CEC2019 functions are known as the 100-digit challenge. These single-factor functions are very appropriate for evaluating other functions because they have totally complicated structures. Some of their features have multiple states, and they are inseparable. Due to space limitation, Table 6 describes these functions briefly. The formulas and mathematical specifications of these functions can be seen in [72] in details. Table 12 shows the results of comparing the algorithms quantitatively. With regard to the mean of results, ICO outperformed other methods in F30 and F35. LASHAD-SPASMA could reach the better result in functions F33 and F39, while GSA could reach the better result in F34 and F38. Also LSHDE reached the better result in functions F30 and F32 and HGSO reached the better result in functions F30 and F31. SASS algorithm had the better results in F36.

By considering Wilcoxon Signed-Rank test results presented in Table 17, it can be said that ICO presents the best results in terms of solving CEC2019 complicated problems except four methods including SASS, LSHADE, LASHAD-SPASMA and LASHAD-cnEspin.

3.2.6. Wilcoxon signed-rank test

In order to show the obtained results of ICO that is significantly different from the other randomized algorithms, the Wilcoxon Signed-Rank Test was used for pairwise comparison. The test was performed by using the global minimum values obtained by performing 30 runs for problem based pairwise comparison of the algorithms. The significance value α was chosen to be 0.05 with null hypothesis H0; that is, there is no difference between the median of the solutions obtained after the same test problem is executed by algorithms A and B, i.e. median (A) = median(B). Also, to determine whether algorithm A yielded statistically the better solution than algorithm B or whether alternative hypothesis was valid, the sizes of the ranks provided by the Wilcoxon Signed-Rank test (T^+ and T^-) were thoroughly examined.

The non-parametric statistical results of the ICO algorithm versus other algorithms based on the Wilcoxon Signed-Rank Test with the statistical significance level $\alpha = 0.05$ are also summarized in Tables 13–17. In these tables, '+' indicated that the null hypothesis H0 was rejected and ICO performed better, while '−' indicated that the null hypothesis H0 was rejected; however, ICO performed worse. The '=' indicates a failure to reject the null hypothesis, and also there is no statistical difference between the two algorithms. The counts of statistical significant cases (+/−/=) are presented in the last row of Tables 13–17.

Table 8
Results of unimodal benchmark functions (F1–F7), with 30 dimensions.

F		ICO	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F1	best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.06E−88	3.50E−66	1.32E−100	1.97E−17	2.21E−09	5.99E−11	5.50E−25	1.62E−227	2.46E−215	**0.00E+00**	3.36E−215	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.00E+03	1.29E−61	5.98E−97	3.07E−17	3.98E−09	2.80E−10	8.81E−03	1.02E−202	3.69E−198	**0.00E+00**	7.53E−194	**0.00E+00**	**0.00E+00**
	std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.84E+03	4.32E−61	1.82E−96	8.01E−18	7.14E−10	2.94E−10	2.67E−02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	best	**0.00E+00**	4.94e−324	**0.00E+00**	**0.00E+00**	4.89E−56	1.70E−42	4.37E−58	2.03E−08	2.46E−05	3.31E+01	4.67E−89	4.73E−101	3.71E−86	6.09E−134	4.02E−98	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	2.96e−323	**0.00E+00**	**0.00E+00**	2.87E+01	1.59E−34	1.79E−56	2.59E−08	2.17E−01	4.40E+01	5.77E−61	1.56E−87	6.49E−69	3.03E−101	2.88E−86	**0.00E+00**	**0.00E+00**
	std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.81E+01	7.39E−34	3.63E−56	3.23E−09	3.46E−01	6.17E+00	3.16E−60	6.70E−87	3.13E−68	1.62E−100	1.18E−85	**0.00E+00**	**0.00E+00**
F3	best	**0.00E+00**	9.08E−276	1.47E−208	8.31E−04	3.73E−10	3.26E−05	1.58E+03	9.82E−17	3.55E−08	1.78E−07	9.04E−16	5.50E−49	3.53E−58	5.14E−280	1.52E−48	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	1.51E−253	3.28E−180	1.77E+01	1.98E+04	4.70E−04	4.39E+03	1.98E−16	5.64E−08	1.71E−06	3.84E−01	1.15E−43	1.75E−49	1.68E−265	3.85E−42	**0.00E+00**	**0.00E+00**
	std	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.64E+01	1.34E+04	4.79E−04	2.78E+03	1.34E−16	1.56E−08	3.57E−06	1.33E+00	5.83E−43	9.11E−49	**0.00E+00**	1.03E−41	**0.00E+00**	**0.00E+00**
F4	best	**0.00E+00**	1.07E−230	7.39E−158	4.76E−17	4.21E+01	6.41E−03	3.18E−15	2.63E−09	3.67E−05	3.26E−05	1.78E+01	4.61E−06	2.68E−06	1.37E−23	4.03E−07	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	1.87E−214	1.64E−151	2.82E+00	6.54E+01	1.92E−02	2.04E−11	3.71E−09	1.93E−01	2.60E−01	2.99E+01	1.16E−04	9.22E−05	2.19E−17	1.29E−05	1.34E−03	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	7.56E−151	9.01E+00	9.72E+00	1.10E−02	7.72E−11	6.40E−10	5.72E−01	4.05E−01	7.32E+00	1.75E−04	1.30E−04	7.50E−17	1.60E−05	7.34E−03	**0.00E+00**
F5	best	2.83E+01	1.70E+01	2.42E+01	2.38E+01	6.10E−02	4.00E+00	4.77E−03	1.43E+01	2.14E+01	2.41E−03	8.22E−02	7.31E−29	6.14E−16	**0.00E+00**	1.16E−28	2.57E+01	2.68E+01
	mean	2.85E+01	1.94E+01	2.62E+01	2.43E+01	2.67E+06	3.86E+01	1.93E+00	1.53E+01	4.47E+01	2.36E+01	1.46E+02	1.33E+00	1.06E+00	1.06E+00	**3.99E−01**	2.66E+01	2.78E+01
	Std	**1.12E−01**	1.26E+00	9.71E−01	5.62E−01	1.46E+07	2.86E+01	1.46E+00	3.76E−01	3.80E+01	2.50E+01	3.10E+02	1.91E+00	1.79E+00	1.79E+00	1.22E+00	5.96E−01	6.46E−01
F6	best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.00E+00	4.60E+01	1.00E+00	1.00E+00	8.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.67E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.37E+00	1.62E+03	7.47E+01	9.93E+00	4.33E+01	1.86E+01	4.40E+00	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.79E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.40E+00	1.73E+03	1.46E+02	1.47E+01	2.40E+01	1.65E+01	5.15E+00	**0.00E+00**	**0.00E+00**
F7	best	1.73E−07	1.67E−05	1.18E−05	5.30E−06	1.85E−02	2.87E−03	1.91E−03	1.75E−02	3.23E−03	1.60E−05	2.40E−03	1.60E−03	1.24E−03	9.73E−04	1.05E−03	**1.43E−07**	2.67E−07
	mean	3.68E−05	6.34E−05	6.45E−05	1.45E−04	3.74E+00	8.67E−03	3.18E−03	2.96E−02	8.48E−03	1.14E−04	1.40E−02	6.57E−03	5.17E−03	7.82E−03	6.13E−03	**4.03E−06**	7.51E−06
	Std	7.31E−05	3.00E−05	3.82E−05	1.55E−04	6.02E+00	3.07E−03	6.62E−04	7.23E−03	4.35E−03	1.19E−04	9.76E−03	4.70E−03	3.23E−03	8.56E−03	5.34E−03	**3.98E−06**	7.19E−06

Table 9
Results of multimodal benchmark functions (F8–F13), with 30 dimensions.

F		ICO	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
	best	−9.90E+03	−1.02E+04	−7.51E+03	−1.26E+04	−1.04E+04	−8.48E+03	−1.25E+04	−4.01E+03	−9.81E+03	−1.26E+04	−1.12E+04	−1.26E+04	−1.36E+04	−1.26E+04	−1.26E+04	−9.6E+03	−**1.3E+28**
F8	mean	−9.02E+03	−9.18E+03	−6.24E+03	−1.23E+04	−8.79E+03	−6.76E+03	−1.22E+04	−2.61E+03	−7.74E+03	−1.22E+04	−1.02E+04	−1.24E+04	−1.26E+04	−1.25E+04	−1.25E+04	−8.4E+03	**−4.3E+26**
	std	4.87E+02	6.93E+02	6.58E+02	5.94E+02	8.69E+02	6.76E+02	2.31E+02	5.05E+02	8.64E+02	2.87E+02	6.00E+02	1.57E+02	5.51E+02	1.05E+02	**6.17E+01**	4.92E+02	2.34E+27
	best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.25E+01	1.39E+01	6.61E+01	1.19E+01	3.98E+01	2.09E+01	1.79E+01	**0.00E+00**	**0.00E+00**	2.98E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
F9	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.79E−15	1.52E+02	2.39E+01	8.57E+01	1.99E+01	6.59E+01	4.36E+01	3.73E+01	6.06E−14	4.97E−01	1.76E+01	6.63E−14	**0.00E+00**	**0.00E+00**
	std	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.08E−14	3.13E+01	6.85E+00	7.34E+00	3.60E+00	2.09E+01	1.18E+01	1.57E+01	3.32E−14	6.27E−01	6.66E+00	3.97E−14	**0.00E+00**	**0.00E+00**
	best	4.41E−11	4.44E−15	4.44E−15	**8.88E−16**	7.99E−15	7.99E−15	4.44E−15	3.62E−09	1.32E−05	3.29E−14	1.34E+00	9.31E−01	1.50E+00	1.51E−14	1.51E−14	**8.88E−16**	**8.88E−16**
F10	mean	8.77E−11	4.44E−15	7.28E−15	3.38E−15	1.34E+01	1.04E−14	9.63E−01	4.34E−09	1.66E+00	2.25E−13	2.96E+00	2.03E+00	3.27E+00	2.05E+00	1.72E+00	**8.88E−16**	1.01E−15
	std	2.32E−11	**0.00E+00**	1.45E−15	1.90E−15	7.92E+00	3.41E−15	3.66E+00	3.65E−10	8.70E−01	4.46E−13	1.22E+00	6.86E−01	8.57E−01	7.37E−01	7.76E−01	**0.00E+00**	6.49E−16
	best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.48E−09	6.67E−13	1.11E−16	**0.00E+00**	2.22E−16	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	mean	**0.00E+00**	**0.00E+00**	3.30E−04	**0.00E+00**	1.81E+01	1.48E−02	**0.00E+00**	1.89E−03	1.61E−02	1.30E−02	1.11E−01	1.31E−02	1.94E−02	3.21E−02	1.53E−02	3.63E−02	**0.00E+00**
	std	**0.00E+00**	**0.00E+00**	1.81E−03	**0.00E+00**	4.37E+01	1.30E−02	**0.00E+00**	3.92E−03	1.47E−02	2.06E−02	2.39E−01	1.34E−02	2.19E−02	3.20E−02	2.53E−02	2.80E−02	**0.00E+00**
	best	1.89E−14	**1.57E−32**	2.90E−03	2.59E−07	2.32E−31	**1.57E−32**	**1.57E−32**	1.45E−19	8.29E−12	1.17E−07	2.14E−10	**1.57E−32**	1.58E−32	**1.57E−32**	**1.57E−32**	6.92E−02	1.28E−01
F12	mean	3.53E−14	4.24E−32	2.88E−02	8.61E−07	8.53E+06	1.64E−32	**1.57E−32**	2.24E−19	1.45E+00	5.01E−07	1.65E+00	2.22E−01	2.50E−01	2.28E−01	1.83E−01	8.96E−02	2.99E−01
	std	9.31E−15	1.11E−31	1.69E−02	3.12E−07	4.67E+07	1.09E−33	**5.57E−48**	7.03E−20	1.78E+00	2.75E−07	1.89E+00	3.02E−01	4.28E−01	5.07E−01	5.25E−01	1.60E−02	1.10E−01
	best	1.98E−13	**1.35E−32**	1.00E−01	5.84E−06	1.98E−30	**1.35E−32**	**1.35E−32**	2.31E−18	1.14E−10	3.29E−08	8.20E−19	**1.35E−32**	**1.35E−32**	**1.35E−32**	**1.35E−32**	2.34E+00	1.41E+00
F13	mean	7.32E−04	3.18E−02	4.33E−01	3.84E−04	2.73E+07	3.66E−04	**1.35E−32**	3.45E−18	3.30E−03	4.26E−07	5.52E−01	1.80E−01	1.84E+00	7.29E−02	1.61E−02	2.65E+00	2.65E+00
	std	2.79E−03	4.27E−02	2.12E−01	2.00E−03	1.04E+08	2.01E−03	**5.57E−48**	1.04E−18	5.12E−03	3.22E−07	8.13E−01	7.08E−01	4.66E+00	2.93E−01	4.30E−02	1.38E−01	3.61E−01

Table 10
Results of fixed-dimension multimodal benchmark functions (F14–F23).

F		ICO	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
	best	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	9.98E−01	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**	**9.98E−01**
F14	mean	1.56E+00	**9.98E−01**	4.36E+00	1.49E+00	1.89E+00	3.55E+00	**9.98E−01**	4.64E+00	1.06E+00	2.69E+00	1.79E+00	**9.98E−01**	**9.98E−01**	1.23E+00	**9.98E−01**	1.04E+01	1.60E+00
	std	8.10E−01	1.30E−16	4.17E+00	1.83E+00	1.41E+00	3.38E+00	**0.00E+00**	3.64E+00	2.52E−01	4.03E+00	1.90E+00	**0.00E+00**	**0.00E+00**	1.09E+00	**0.00E+00**	3.77E+00	5.53E−01
	best	3.13E−04	**3.07E−04**	**3.07E−04**	3.08E−04	5.75E−04	3.10E−04	**3.07E−04**	1.16E−03	3.45E−04	3.08E−04	**3.07E−04**	**3.07E−04**	**3.07E−04**	**3.07E−04**	**3.07E−04**	3.29E−04	3.17E−04
F15	mean	5.23E−04	3.01E−03	5.56E−03	5.66E−04	8.21E−04	7.95E−04	**3.07E−04**	2.59E−03	1.41E−03	1.55E−03	3.82E−04	1.71E−03	1.23E−03	3.11E−04	9.76E−04	7.65E−03	3.56E−04
	std	1.40E−04	6.92E−03	1.22E−02	3.24E−04	3.24E−04	1.99E−04	**1.50E−13**	1.89E−03	3.59E−03	4.00E−03	2.01E−04	5.08E−03	3.33E−03	2.13E−05	3.66E−03	1.17E−02	3.04E−05
	best	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
F16	mean	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	−1.03E+00
	std	4.12E−13	6.12E−16	1.33E−08	3.39E−10	6.78E−16	6.39E−16	6.78E−16	5.05E−16	1.14E−14	**4.61E−16**	6.78E−16	6.78E−16	6.78E−16	6.71E−16	6.78E−16	1.42E−07	8.26E−05
	best	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	3.98E−01	3.98E−01
F17	mean	**3.98E−01**	**3.98E−01**	3.98E−01	3.98E−01	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	4.09E−01	3.99E−01
	std	1.96E−13	**0.00E+00**	1.57E−06	4.44E−06	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.45E−14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.13E−03	9.83E−04
	best	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.00E+00
F18	mean	**3.00E+00**	**3.00E+00**	3.00E+00	3.00E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	1.13E+01	3.00E+00
	std	4.43E−12	2.00E−15	3.74E−05	7.48E−05	1.97E−14	1.42E−14	2.08E−15	1.39E−14	3.15E−13	3.44E−14	1.42E−14	1.45E−14	1.91E−15	**1.33E−15**	1.45E−14	1.25E+01	1.28E−04
	best	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	−3.86E+00	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	−3.86E+00	−3.86E+00
F19	mean	**−3.86E+00**	−3.86E+00	−3.86E+00	−3.86E+00	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	−3.85E+00	−3.86E+00
	std	6.84E−13	1.44E−03	2.49E−03	2.60E−03	2.71E−15	2.70E−15	2.71E−15	**2.36E−15**	3.83E−14	2.70E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	4.91E−03	3.18E−03
	best	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	−3.32E+00	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	−3.24E+00	−3.24E+00
F20	mean	−3.32E+00	−3.27E+00	−3.29E+00	−3.29E+00	−3.24E+00	−3.29E+00	−3.29E+00	**−3.32E+00**	−3.24E+00	−3.27E+00	−3.27E+00	−3.27E+00	−3.29E+00	−3.28E+00	−3.31E+00	−3.15E+00	−3.13E+00
	std	2.17E−02	6.03E−02	6.14E−02	6.08E−02	6.76E−02	5.35E−02	5.54E−02	**1.37E−15**	5.55E−02	5.92E−02	5.99E−02	5.92E−02	5.54E−02	5.70E−02	4.11E−02	5.30E−02	7.74E−02
	best	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	**−1.02E+01**	−6.28E+00	−5.03E+00
F21	mean	−8.97E+00	−9.05E+00	−8.97E+00	−9.64E+00	−7.22E+00	−7.08E+00	**−9.90E+00**	−6.07E+00	−8.81E+00	−5.24E+00	−8.65E+00	−7.90E+00	−8.48E+00	−9.13E+00	−9.57E+00	−3.93E+00	−4.89E+00
	std	2.18E+00	2.27E+00	2.19E+00	1.55E+00	3.30E+00	2.96E+00	1.36E+00	3.50E+00	2.54E+00	3.37E+00	2.83E+00	3.11E+00	2.89E+00	2.07E+00	1.82E+00	1.16E+00	**1.01E−01**
	best	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	−8.00E+00	−5.03E+00
F22	mean	−1.01E+01	−9.69E+00	−1.02E+01	−9.12E+00	−7.98E+00	−1.01E+01	**−1.04E+01**	**−1.04E+01**	−9.97E+00	−6.72E+00	−9.78E+00	−9.75E+00	−9.89E+00	−9.87E+00	−1.01E+01	−4.14E+00	−4.87E+00
	std	1.34E+00	1.84E+00	9.70E−01	2.70E+00	3.31E+00	1.34E+00	1.36E−15	**1.04E−15**	1.67E+00	3.78E+00	1.91E+00	2.02E+00	1.94E+00	1.62E+00	1.39E+00	1.70E+00	1.04E−01
	best	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	−1.05E+01	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	−9.09E+00	−5.11E+00
F23	mean	−1.04E+01	−1.04E+01	−9.91E+00	−9.55E+00	−7.29E+00	−1.03E+01	**−1.05E+01**	−1.04E+01	−1.01E+01	−6.71E+00	−1.00E+01	−9.83E+00	−1.03E+01	−1.04E+01	−1.03E+01	−3.77E+00	−4.96E+00
	std	9.79E−01	9.87E−01	1.97E+00	2.26E+00	3.81E+00	1.22E+00	**1.78E−15**	7.96E−01	1.69E+00	3.91E+00	1.89E+00	2.15E+00	1.48E+00	9.87E−01	1.48E+00	1.88E+00	8.72E−02

Table 11
Results of composite benchmark functions (F24–F29).

F		ICO	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F24	best	5.39E−13	**0.00E+00**	1.26E−01	4.18E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.53E−17	4.30E−11	1.41E−16	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.84E+01	3.79E+01
	mean	1.44E−12	7.00E+01	8.98E+01	1.30E+02	1.02E+02	6.00E+01	1.00E+01	**2.75E−17**	5.33E+01	1.13E+02	2.00E+01	4.33E+01	3.33E+01	4.33E+01	1.33E+01	2.45E+02	1.07E+02
	std	5.19E−13	8.77E+01	1.11E+02	1.18E+02	6.33E+01	7.70E+01	3.05E+01	**9.94E−18**	7.30E+01	1.04E+02	4.84E+01	6.26E+01	6.06E+01	7.74E+01	3.46E+01	1.29E+02	4.40E+01
F25	best	1.36E−11	**0.00E+00**	9.22E+00	1.47E+01	5.60E+00	3.30E+00	**0.00E+00**	1.00E+02	1.07E+01	1.40E+01	1.47E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.85E+02	9.70E+01
	mean	**2.24E+00**	1.26E+02	1.89E+02	1.67E+02	5.09E+01	5.38E+01	6.78E+00	1.86E+02	2.73E+01	2.48E+02	6.75E+01	1.03E+02	9.90E+01	9.11E+01	5.85E+01	5.41E+02	2.75E+02
	std	**1.55E+00**	1.19E+02	1.05E+02	1.09E+02	4.76E+01	8.02E+01	2.25E+01	4.30E+01	1.31E+01	1.39E+02	7.62E+01	1.04E+02	8.01E+01	8.33E+01	8.11E+01	1.90E+02	7.20E+01
F26	best	7.99E−12	6.34E+01	1.10E+02	2.20E+02	1.46E+02	9.69E+01	9.23E+01	**0.00E+00**	1.08E+02	1.07E+02	9.65E+01	6.31E+01	6.72E+01	9.04E+01	6.72E+01	2.99E+02	3.01E+02
	mean	1.21E+02	1.61E+02	2.08E+02	4.24E+02	2.64E+02	1.62E+02	1.23E+02	1.19E+02	1.89E+02	3.74E+02	1.54E+02	1.25E+02	1.40E+02	1.39E+02	**1.17E+02**	6.29E+02	4.38E+02
	std	4.69E+01	4.72E+01	1.06E+02	1.40E+02	8.07E+01	3.87E+01	**2.04E+01**	6.55E+01	4.66E+01	1.43E+02	3.49E+01	5.09E+01	5.43E+01	3.35E+01	5.33E+01	1.68E+02	9.00E+01
F27	best	**2.00E+02**	2.63E+02	2.65E+02	3.58E+02	2.99E+02	2.72E+02	2.96E+02	**2.00E+02**	2.60E+02	3.21E+02	**2.00E+02**	2.28E+02	2.40E+02	2.63E+02	2.52E+02	5.26E+02	3.86E+02
	mean	**2.44E+02**	3.52E+02	3.71E+02	5.16E+02	3.46E+02	3.21E+02	3.08E+02	4.66E+02	3.12E+02	4.73E+02	2.85E+02	3.19E+02	2.97E+02	3.29E+02	2.91E+02	7.77E+02	4.91E+02
	std	2.79E+01	1.25E+02	1.11E+02	1.29E+02	3.77E+01	3.38E+01	**6.21E+00**	1.71E+02	2.47E+01	1.15E+02	3.36E+01	1.02E+02	4.01E+01	9.69E+01	5.93E+01	1.23E+02	5.94E+01
F28	best	6.08E−12	**0.00E+00**	3.20E+00	2.63E+01	3.89E+00	2.13E+00	**0.00E+00**	3.94E−15	2.20E−09	5.13E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.29E+02	3.74E+01
	mean	**3.31E−01**	7.18E+01	1.38E+02	1.39E+02	7.55E+01	8.98E+01	1.73E+01	1.89E+02	2.71E+01	1.53E+02	3.83E+01	5.12E+01	4.47E+01	3.52E+01	4.44E+01	5.38E+02	1.56E+02
	std	**6.19E−01**	1.26E+02	1.49E+02	8.96E+01	7.07E+01	9.72E+01	3.78E+01	8.20E+01	4.15E+01	1.88E+02	4.85E+01	5.03E+01	6.49E+01	5.64E+01	5.00E+01	1.92E+02	6.16E+01
F29	best	**4.00E+02**	4.00E+02	5.01E+02	5.03E+02	5.00E+02	4.02E+02	5.00E+02	5.63E+02	4.02E+02	5.01E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02	**4.00E+02**	5.84E+02	4.79E+02
	mean	**4.77E+02**	8.29E+02	8.90E+02	7.60E+02	7.33E+02	7.35E+02	5.38E+02	7.06E+02	5.35E+02	8.59E+02	6.98E+02	7.92E+02	7.85E+02	7.41E+02	6.95E+02	8.80E+02	7.19E+02
	std	**4.30E+01**	1.60E+02	7.35E+01	1.92E+02	1.89E+02	2.02E+02	1.13E+02	7.98E+01	1.32E+02	1.21E+02	2.02E+02	1.80E+02	1.77E+02	2.00E+02	2.06E+02	6.75E+01	1.74E+02

Table 12
Results of CEC2019 functions (F30–F39).

F		ICO	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
	best	**1.00E+00**	**1.00E+00**	**1.00E+00**	3.47E+02	5.33E+04	1.30E+02	2.33E+04	3.85E+07	4.60E+04	2.35E+03	3.50E+01	**1.00E+00**	**1.00E+00**	1.37E+01	**1.00E+00**	**1.00E+00**	**1.00E+00**
F30	mean	**1.00E+00**	7.28E+02	6.98E+02	1.76E+06	1.06E+07	2.72E+07	7.67E+04	1.63E+08	3.76E+05	4.59E+04	3.16E+04	**1.00E+00**	1.17E+00	2.69E+08	**1.00E+00**	**1.00E+00**	**1.00E+00**
	std	1.01E−13	3.47E+03	2.71E+03	3.00E+06	2.34E+07	4.27E+07	4.26E+04	1.11E+08	3.23E+05	4.83E+04	4.33E+04	3.87E−11	3.54E−01	2.57E+08	**0.00E+00**	3.67E−09	9.84E−13
	best	4.17E+00	4.26E+00	4.37E+00	2.07E+03	3.14E+02	1.29E+03	1.35E+03	6.03E+03	6.16E+01	8.11E+01	3.28E+01	8.29E+00	1.55E+01	**3.30E+00**	5.39E+00	4.16E+01	4.22E+00
F31	mean	4.57E+00	4.80E+01	1.80E+02	6.49E+03	1.83E+03	4.45E+03	2.02E+03	1.80E+04	4.54E+02	2.77E+02	1.87E+02	5.30E+01	1.18E+02	1.15E+02	4.73E+01	4.95E+03	**4.32E+00**
	std	7.62E−01	6.93E+01	1.47E+02	2.51E+03	2.94E+03	1.96E+03	3.64E+02	5.96E+03	3.39E+02	2.21E+02	9.52E+01	4.73E+01	6.72E+01	7.84E+01	4.16E+01	2.10E+03	**2.31E−01**
	best	**1.00E+00**	**1.00E+00**	1.00E+00	1.00E+00	1.41E+00	1.41E+00	4.40E+00	1.41E+00	1.41E+00	1.41E+00	**1.00E+00**	1.00E+00	**1.00E+00**	**1.00E+00**	1.00E+00	6.58E+00	4.41E+00
F32	mean	1.33E+00	1.40E+00	2.01E+00	2.69E+00	6.23E+00	2.54E+00	6.04E+00	3.41E+00	2.91E+00	6.32E+00	1.38E+00	**1.25E+00**	1.29E+00	1.31E+00	1.29E+00	8.57E+00	6.24E+00
	std	1.66E−01	**7.47E−02**	1.65E+00	1.40E+00	2.01E+00	2.30E+00	9.78E−01	2.15E+00	2.05E+00	3.32E+00	1.04E−01	2.04E−01	1.91E−01	1.76E−01	1.91E−01	1.12E+00	9.61E−01
	best	2.99E+00	2.99E+00	4.40E+00	2.39E+01	7.96E+00	1.99E+01	8.33E+00	1.99E+01	4.98E+00	7.96E+00	4.98E+00	**1.99E+00**	2.99E+00	1.99E+00	2.99E+00	1.79E+01	3.55E+01
F33	mean	6.87E+00	9.46E+00	1.51E+01	5.16E+01	2.79E+01	3.81E+01	1.23E+01	3.74E+01	1.98E+01	3.07E+01	1.20E+01	5.34E+00	5.05E+00	**4.49E+00**	5.25E+00	4.81E+01	4.64E+01
	std	2.44E+00	3.96E+00	6.44E+00	1.49E+01	1.11E+01	1.22E+01	1.91E+00	8.16E+00	9.47E+00	1.32E+01	5.40E+00	2.16E+00	**1.20E+00**	1.92E+00	1.48E+00	1.71E+01	6.67E+00
	best	1.01E+00	1.01E+00	1.12E+00	1.30E+00	1.04E+00	1.06E+00	1.09E+00	**1.00E+00**	1.08E+00	1.06E+00	1.02E+00	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	8.70E+00	3.84E+00
F34	mean	1.04E+00	1.05E+00	1.51E+00	1.73E+00	2.27E+00	1.13E+00	1.20E+00	**1.01E+00**	1.30E+00	1.37E+00	1.08E+00	1.03E+00	1.02E+00	1.03E+00	1.02E+00	3.08E+01	6.62E+00
	std	2.22E−02	2.91E−02	4.91E−01	2.56E−01	3.51E+00	8.65E−02	5.81E−02	**7.46E−03**	1.37E−01	2.45E−01	4.82E−02	2.23E−02	1.23E−02	1.95E−02	1.56E−02	1.37E+01	1.81E+00
	best	1.00E+00	**1.00E+00**	1.20E+00	3.82E+00	2.78E+00	1.00E+00	**1.00E+00**	**1.00E+00**	1.31E+00	2.09E+00	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	7.76E+00	4.01E+00
F35	mean	**1.00E+00**	1.95E+00	2.37E+00	7.82E+00	4.75E+00	3.37E+00	1.37E+00	2.56E+00	4.28E+00	4.97E+00	1.17E+00	1.04E+00	1.08E+00	1.05E+00	1.03E+00	1.03E+01	6.31E+00
	std	**3.51E−06**	9.83E−01	8.05E−01	1.91E+00	1.53E+00	1.36E+00	3.04E−01	1.29E+00	1.60E+00	1.54E+00	3.68E−01	1.22E−01	3.35E−01	1.13E−01	1.17E−01	1.29E+00	7.96E−01
	best	1.20E+02	1.11E+01	2.74E+02	5.75E+02	4.46E+02	6.53E+02	1.68E+01	1.01E+03	2.38E+02	5.39E+02	7.89E+00	1.96E+01	**1.26E+00**	1.01E+01	1.38E+00	5.84E+02	9.59E+02
F36	mean	3.53E+02	5.03E+02	7.21E+02	1.20E+03	9.65E+02	1.11E+03	3.10E+02	1.56E+03	8.08E+02	9.74E+02	6.97E+02	2.37E+02	2.51E+02	2.35E+02	**2.05E+02**	1.23E+03	1.39E+03
	std	1.78E+02	2.46E+02	2.29E+02	3.14E+02	3.04E+02	2.54E+02	1.38E+02	3.12E+02	2.70E+02	3.03E+02	2.86E+02	**9.74E+01**	1.46E+02	1.32E+02	1.32E+02	3.13E+02	1.62E+02
	best	2.54E+00	1.50E+00	1.92E+00	4.05E+00	3.43E+00	3.21E+00	3.22E+00	4.65E+00	2.64E+00	3.69E+00	2.56E+00	2.01E+00	**1.35E+00**	1.69E+00	1.41E+00	3.93E+00	3.72E+00
F37	mean	3.33E+00	3.06E+00	3.40E+00	4.53E+00	4.39E+00	4.10E+00	3.64E+00	5.19E+00	3.81E+00	4.31E+00	3.62E+00	2.85E+00	**2.49E+00**	2.57E+00	2.72E+00	4.76E+00	4.32E+00
	std	4.79E−01	6.09E−01	6.08E−01	3.14E−01	4.10E−01	4.29E−01	2.22E−01	2.38E−01	4.89E−01	3.08E−01	4.92E−01	3.66E−01	4.11E−01	4.24E−01	4.51E−01	2.98E−01	**2.06E−01**
	best	1.02E+00	1.02E+00	1.06E+00	1.10E+00	1.08E+00	1.07E+00	1.09E+00	**1.01E+00**	1.08E+00	1.05E+00	1.02E+00	1.02E+00	1.02E+00	1.02E+00	1.02E+00	1.17E+00	1.26E+00
F38	mean	1.04E+00	1.06E+00	1.14E+00	1.36E+00	1.38E+00	1.18E+00	1.17E+00	**1.04E+00**	1.22E+00	1.33E+00	1.09E+00	1.08E+00	1.06E+00	1.07E+00	1.10E+00	1.31E+00	1.45E+00
	std	1.87E−02	3.97E−02	5.70E−02	1.85E−01	1.92E−01	7.86E−02	3.14E−02	**1.31E−02**	1.28E−01	1.69E−01	4.82E−02	3.12E−02	3.44E−02	3.22E−02	3.73E−02	1.40E−01	8.93E−02
	best	**1.00E+00**	**1.00E+00**	2.12E+01	2.10E+01	2.10E+01	**1.00E+00**	2.11E+01	**1.00E+00**	2.10E+01	**1.00E+00**	2.10E+01	1.12E+00	**1.00E+00**	**1.00E+00**	1.00E+00	2.06E+01	1.33E+01
F39	mean	1.71E+01	1.91E+01	2.13E+01	2.11E+01	2.12E+01	2.04E+01	2.11E+01	1.37E+01	2.10E+01	2.03E+01	2.10E+01	1.65E+01	1.46E+01	**1.06E+01**	1.66E+01	2.10E+01	2.00E+01
	std	7.51E+00	6.02E+00	7.58E−02	9.54E−02	1.47E−01	3.67E+00	**4.22E−02**	9.80E+00	4.52E−02	3.65E+00	5.04E−02	7.85E+00	9.25E+00	9.32E+00	8.03E+00	9.13E−02	2.27E+00

The Wilcoxon Signed-Rank Test results for each test category are presented in Table 18. In this table, each cell shows total count of three statistical significance cases ($+/=/-$) in the pairwise comparison. The results show that ICO can achieve statistically better results than comparison algorithms except SASS, with a level of significance $\alpha = 0.05$. However, the numerical results provided for some algorithms such as DE in fixed-dimension multimodal functions is not sufficient to make a statistically convincing inference. With regard to Wilcoxon Signed-Rank Test in Table 18, the performance success of ICO in fixed-dimension multimodal functions is relatively low compared to others. This fits "No-Free-Lunch" (NFL) theorem.

3.2.7. Friedman ranking (FR) test

The Friedman test was employed to compare the proposed method with other state-of-art algorithms presented in this paper. It is a nonparametric test and equivalent to the analysis of variance with repetitive sizes (within groups). This test is utilized to compare mean of ranks between k variables (groups). According to the ranking performed through the Friedman test, the proposed method was ranked as the highest in all existing methods by considering 39 different test functions. Table 19 shows the rank of each method. The following equation can be employed to determine the superiority of ICO in other methods:

$$(\text{new\_value} - \text{original\_value})/|\text{original\_value}| \times 100 \tag{24}$$

Where new_value shows the rank of ICO when original_value indicates the rank of each technique used to calculate the superiority of ICO to that method. According to the results, ICO could outperform EO, GWO, WOA, MFO, PSO, DE, GSA, SSA, SHO, GA-MPC, LSHADE, LSHADE-cnEpSin, LSHADE-SPACMA, SASS, AOA and HGSO 28%, 47%, 53%, 61%, 47%, 21%, 44%, 52%, 56%, 47%, 31%, 27%, 22%, 7%, 61% and 52% respectively in 39 benchmark functions. The P-value of Friedman rank test was reported 7.2868e−25. Since it is smaller than 0.05, it can be concluded that the results of different algorithms are significantly different.

3.3. Convergence analysis

In this part, the convergence of ICO algorithm is investigated. Some benchmark functions are evaluated in this step. In this step of evaluations, the number of initial population is 30, and MaxFEs is considered as $D \times 10{,}000$. In Fig. 4, trajectory in the 1st dimension, search history, the mean of solution fitness and convergence diagrams are demonstrated. The first column of Fig. 4, shows 2D representations of benchmark mathematical functions. The second column of Fig. 4, search history, involves the solution concentration from the first iteration to the last iteration. In order to understand that how solutions can explore and exploit the domain, the concentration is demonstrated on the counter lines of search space. Also the search history can clarify the pattern used by the solution to search the space. In evaluation algorithm, solutions extensively tend to search promising regions of the search spaces and exploit the optimum point. This pattern can be observed in unimodal, multimodal, fixed-dimension multimodal and composite functions and CEC2019. These results of ICO algorithm creates a balance between exploration and exploitation effectively so that the solutions move toward the optimum point.

In addition, the third column of Fig. 4 shows the variation of concentration in the first dimension of the first solution. In the initial steps, the iteration of sudden changes is observed, and it gradually decreases during the iteration. According to Berg and et al. [73], this behavior can guarantee that an algorithm can converge in a point of search space. In order to confirm that this behavior results in solutions fitness, the mean of solutions fitness and convergence diagram is respectively presented in the fourth and fifth columns in Fig. 4. The mean of solutions fitness presents an attitude about the total behavior of all solutions in optimization process. The diagrams have descending behavior in all test functions. It proves that ICO improves the precision of approximated optimum considerably in all iterations. Figs. 5–7 presents the convergence diagram of the proposed method in comparison with different algorithms in 39 benchmark functions. In unimodal functions, the proposed method is considered as one of the top 4 methods in terms of convergence in all functions except F5 among 17 methods. The convergence of the proposed method is faster than other methods in composite functions. CEC2019 is better than some methods, but it has weaker performance than some other methods. ICO has the weaker performance in multimodal functions, and it has less convergence speed compared to most of methods. According to NFL theory, an algorithm cannot be the best one in all modes, and this is also true for ICO. The proposed method has the better results in the problems that are close to the real world like composite functions.

Totally, although convergence speed of the proposed algorithm is less in most functions in the beginning of searching, after some iterations, it performs better than other algorithms in terms of convergence, and the response is better than other methods in most functions especially composite functions that are close to the real problems.

3.4. Scalability analysis

Real-world optimization problems almost involve many variables. In this part, by analyzing scalability, the performance and efficiency of ICO algorithm is investigated in terms of solving the problems with the higher dimensions. This test is studied in various dimensions by considering the fix number of initial population (30), Smax (20) and MaxFEs (le5).

The algorithm starts from 100 to 1000 dimensions with a step size of 100. The performance of algorithm with the fix values of N, Smax and MaxFEs is shown by this test. The evaluations are performed on some samples of unimodal and multimodal functions, and it is compared with other known methods including EO, GWO and PSO. The evaluation results of unimodal functions (F1, F3 and F4) and multimodal functions (F9, F10 and F11) are presented in Fig. 8.

ICO algorithm could reach the global optimum point (0) with high performance in functions F1, F9 and F11. Therefore, they are not shown in these diagrams. Like ICO, EO algorithm could reach the global optimum point in all dimensions in function F9, but it could reach that point in dimension smaller than 800 in function F11. GWO could reach this point in function F9 in dimension 100 and in function F11 in dimensions 100 and 200.

In function F3, the results of ICO are very close to the global optimum in all dimensions, and it has the better performance than PSO, GWO and EO algorithms. In functions F10 and F4, ICO has the better performance in different dimensions in comparison to other methods.

In order to compare convergence diagram of ICO exactly with other methods of PSO, GWO and EO, they are demonstrated in Fig. 9 with dimensions 10, 100 and 1000. As it is obvious in the figure, by increasing dimensions, the performance of PSO, GWO and EO algorithms decreases. But, In ICO method, by increasing dimensions, trivial reduction is observed in its precision. It shows its higher scalability of the proposed method in compared to other methods.

3.5. Computational time analysis

Another evaluation criterion used to compare the algorithms' performance is to compare their execution time in terms of solving different problems. The mean sum of 30 different runs

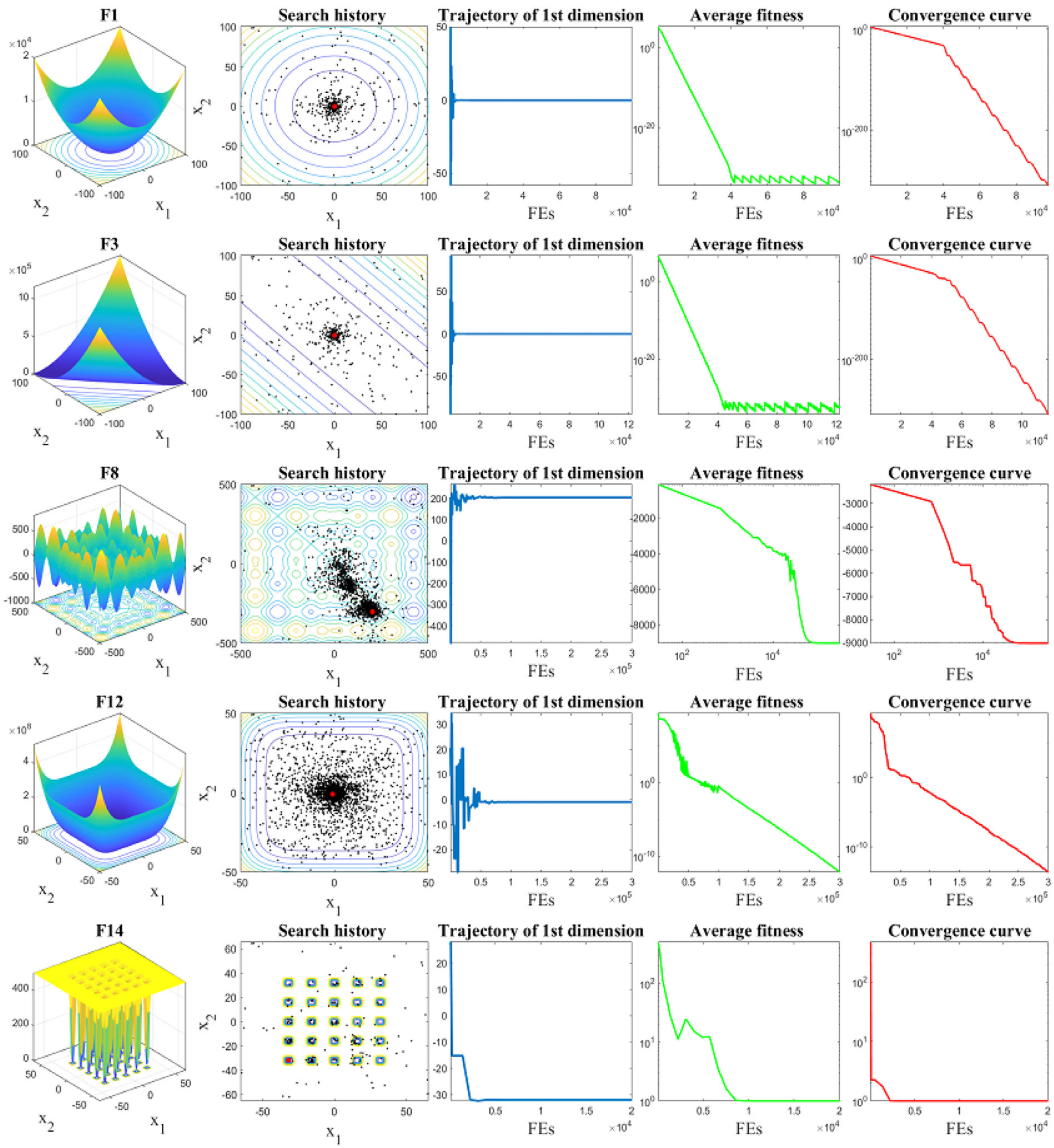

Fig. 4. Search history, optimization history, average fitness history and trajectory in 1st dimension.

in each algorithm is shown in Fig. 10 to solve 39 functions (unimodal, multimodal, fixed dimension multimodal, composite functions and CEC2019 functions). According to the figure, execution time of ICO is less than execution time of known and efficient algorithms like DE, LASHADE-SPACMA, LSHADE-cnEpSin and HGSO.

4. ICO in engineering problems

The proposed method was adopted to solve three famous engineering problems, i.e. pressure vessel design, welded beam design, and Tension/Compression Spring design. The efficiency of this method was then evaluated in terms of solving these problems. A simple method of limitation control called static penalty is also used to solve these problems to greatly penalize the target performance if the limitation of any predefined boundaries is violated. The penalty coefficient should be large enough to penalize the target performance in equality/inequality limitations.

The following subsections present the relationships in each problem as well as the results of using ICO to solve them. Moreover, all of the evaluations were repeated for 30 times by considering 5 initial populations and $D \times 10{,}000$ MaxFEs. The corresponding tables show the mean of results, the best result, the worst result, and the standard deviation in comparison with other methods.

4.1. Pressure vessel design

The goal of this problem is to minimize the total cost of procuring materials, forming, and welding a cylindrical vessel

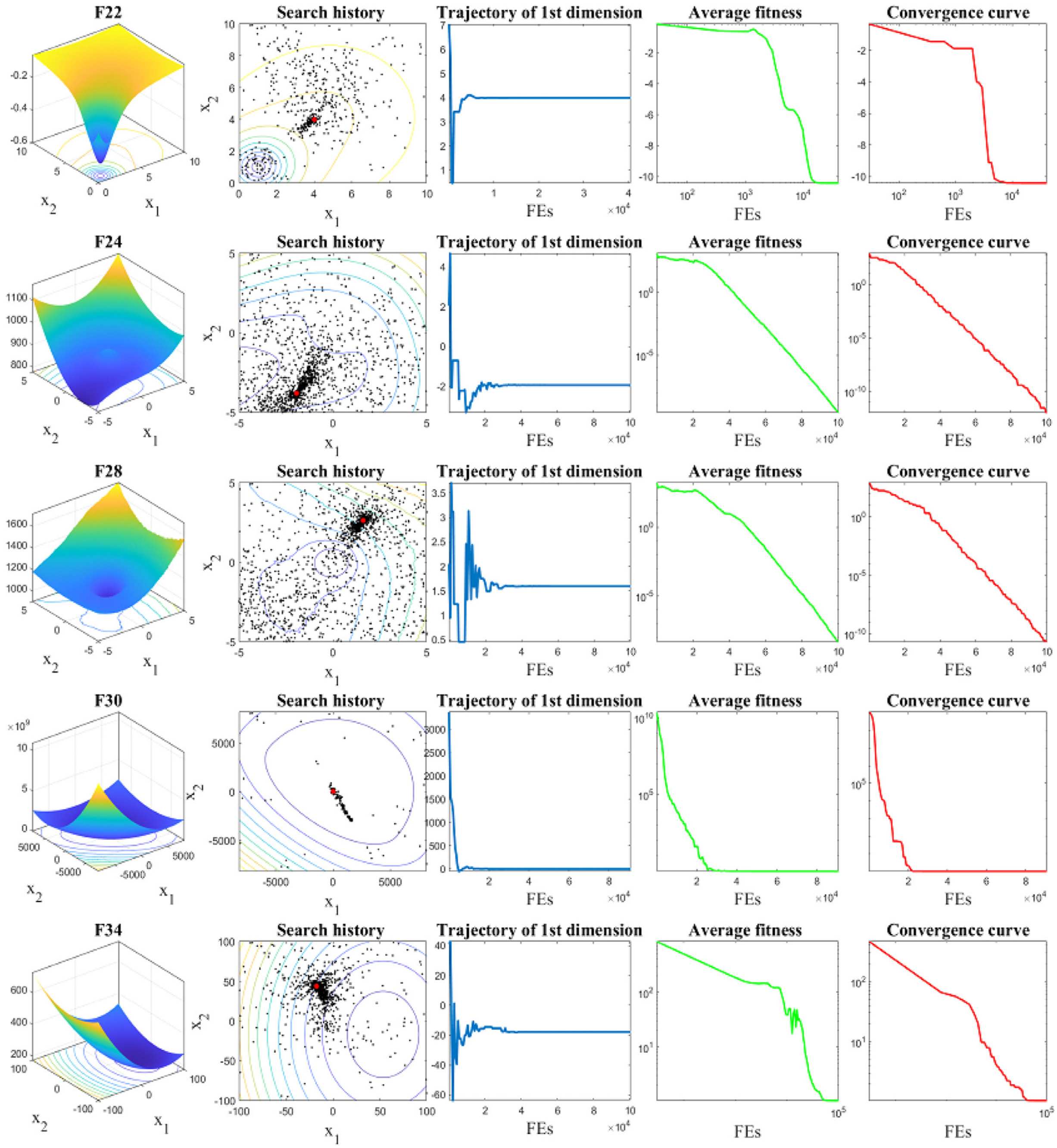

Fig. 4. (*continued*).

shown in Fig. 11. There are four design variables in this problem including x_1 (T_s, shell thickness), x_2 (T_h, nose thickness), x_3 (R, internal radius), and x_4 (L, length of the cylindrical segment). In this problem, T_s and T_h are correct multiples of 0.0625 inch, that is the available thickness of rolled steel plates. In addition, R and L are continuous parameters. The problem is defined as follows:

Minimization of

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_3$$

For

$$\begin{aligned} g_1(x) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(x) &= -x_2 + 0.0954x_3 \le 0 \\ g_3(x) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(x) &= x_4 - 240 \le 0 \end{aligned} \tag{25}$$

The following ranges are used for variables:

$$1 \le x_1 \le 99,\ 1 \le x_2 \le 99,$$
$$10.0 \le x_3 \le 200.0,\ 10.0 \le x_4 \le 200.0$$

Many evolutionary algorithms such as the GSA, MSCA, PRO, SMO, WOA, GWO, PSO and GA were employed to solve this problem. The GSA, MSCA, PRO, and WOA solved this problem with minimum costs of 8538.836, 5935.716, 6050.713 and 6059.741 respectively. By contrary, ICO could solve it with minimum cost of 5891.383032027. In fact, it outperformed not only the above-mentioned methods but also all of the other methods mentioned in Table 20. Moreover, Table 21 shows the mean, maximum, minimum and standard deviation of evaluation results along with those of other methods. Accordingly, ICO outperformed all of these methods.

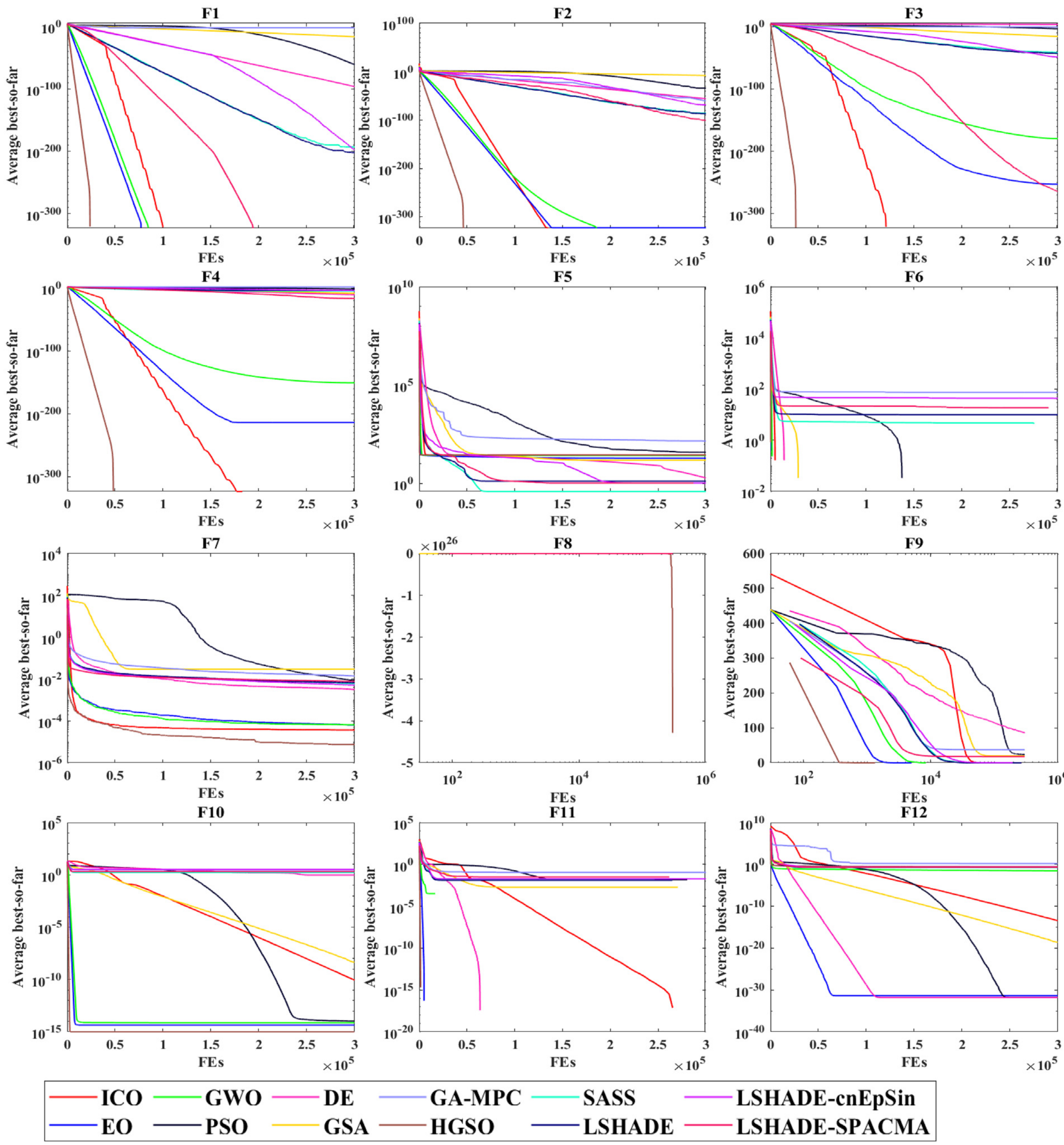

Fig. 5. Convergence curves of the algorithms on F1–F12.

4.2. Welded beam design

The goal of this problem is to minimize the cost of the welded beam with respect to certain constraints on tensile stress, bend stress beam, counter load on the rod and final bend beam as well as side constraints (Fig. 12). The problem is defined as follows:

Minimization of

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4\left(14.0 + x_2\right)$$

For

$$\begin{aligned} &g_1(x) = \tau(X) - \tau_{\max} \le 0 \\ &g_2(x) = \sigma(X) - \sigma_{\max} \le 0 \\ &g_3(x) = x_1 - x_4 \le 0 \\ &g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4\left(14.0 + x_2\right) - 5.0 \le 0 \\ &g_5(x) = 0.125 - x_1 \le 0 \\ &g_6(x) = \delta(x) - \delta_{\max} \le 0 \\ &g_7(x) = P - P_c(x) \end{aligned} \tag{26}$$

With respect to

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

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

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

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

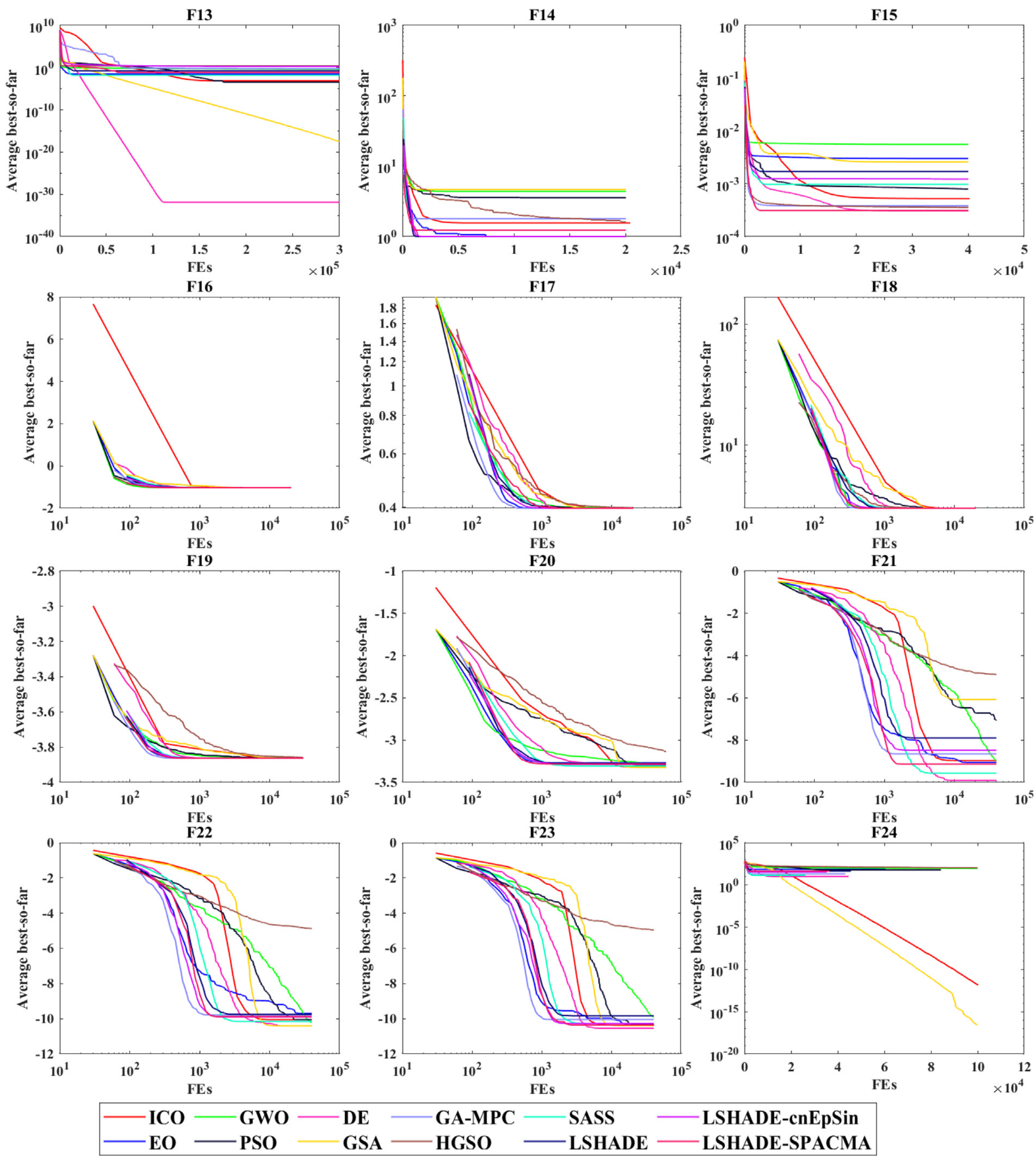

Fig. 6. Convergence curves of the algorithms on F13–F24.

And

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

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

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

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

The following ranges were used for the variables:

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

Many evolutionary algorithms are employed to solve this problem. Table 22 shows the minimum of results and calculated values for variables in comparison with those of other methods. In addition, Table 23 indicates the mean, maximum, minimum and standard deviation of the results obtained from 30 executions. Accordingly, ICO failed to perform well in terms of solving this problem. On the contrary, MSCA outperformed other methods with a minimum of 1.6979.

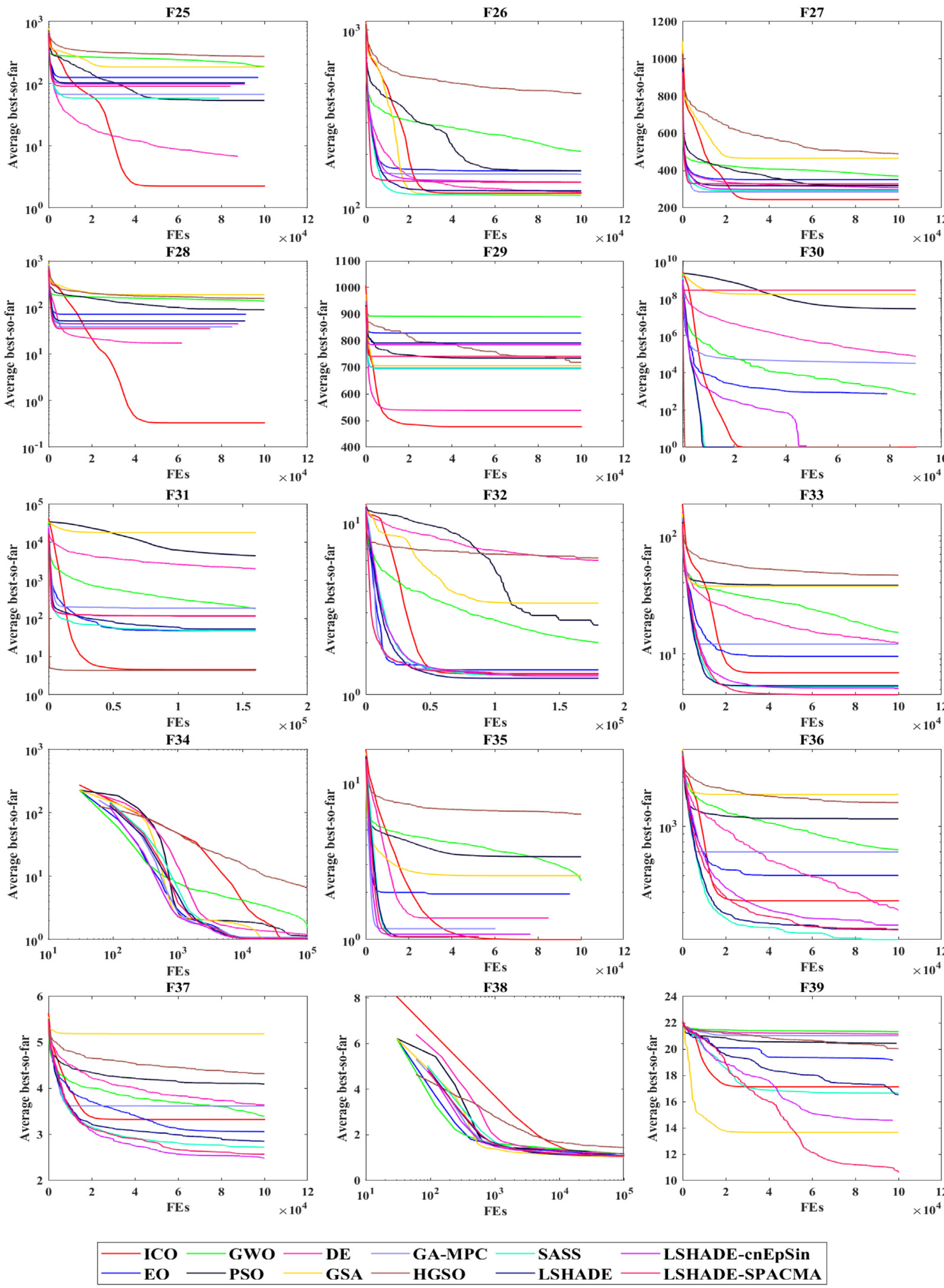

Fig. 7. Convergence curves of the algorithms on F25–F39.

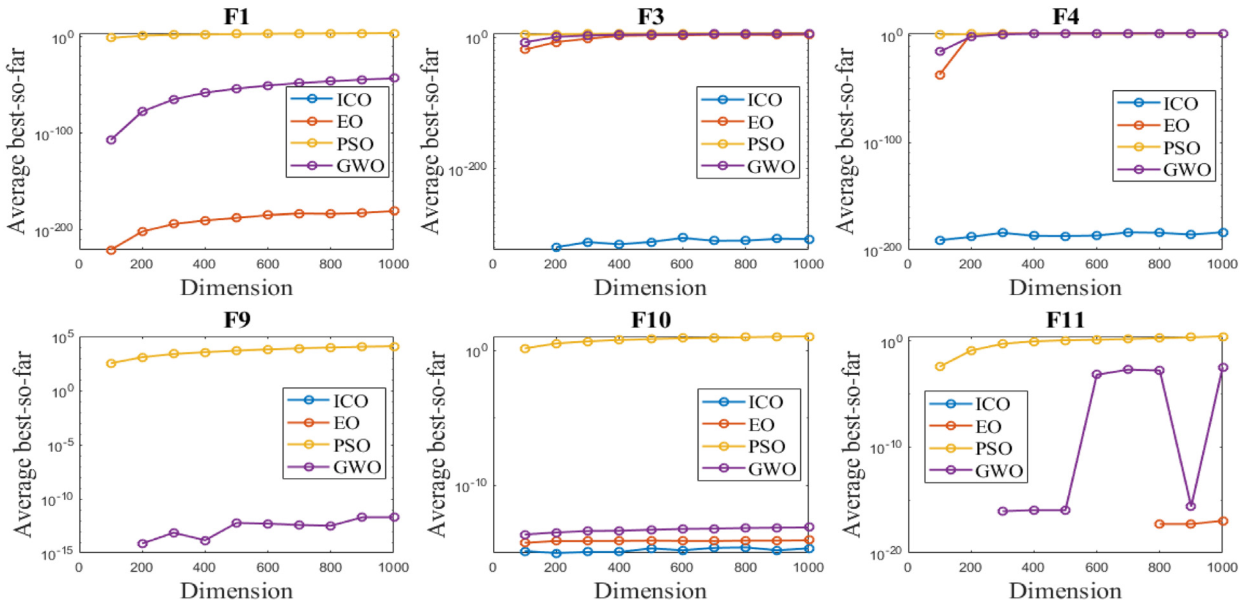

Fig. 8. Robustness comparison of different algorithms in high and very high dimensional test cases.

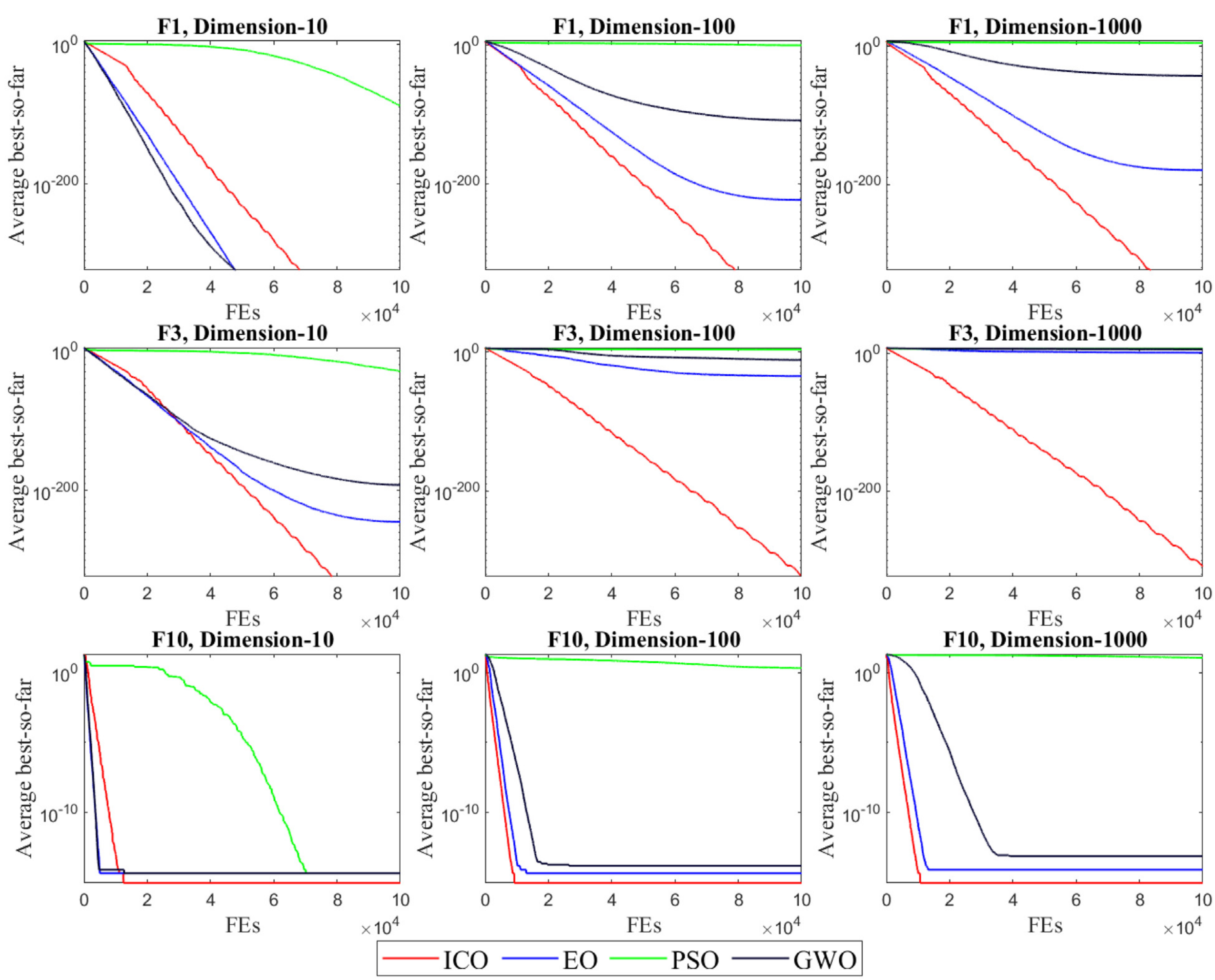

Fig. 9. Convergence of solutions of ICO, EO, GWO, and PSO with respect to FEs in low, high and very high dimensional test cases.

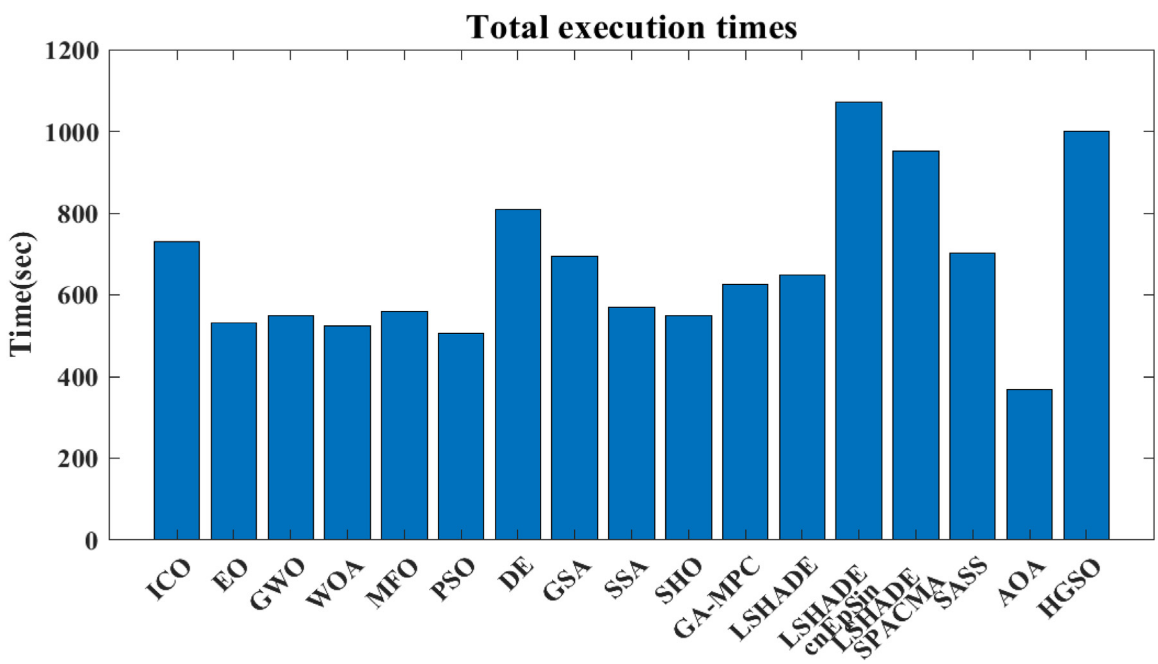

Fig. 10. Total execution time for per algorithm.

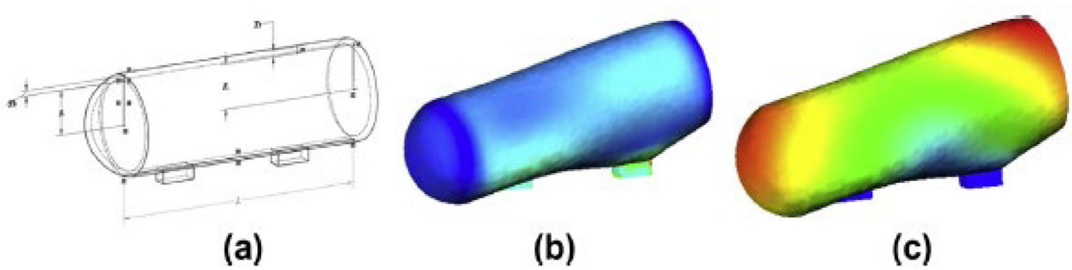

Fig. 11. Pressure vessel (a) schematic (b) stress heatmap (c) displacement heatmap [8].

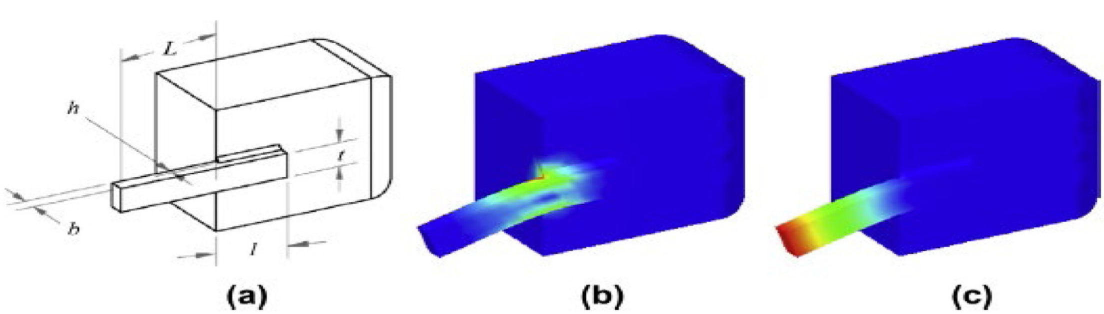

Fig. 12. Structure of welded beam design (a) schematic (b) stress heatmap (c) displacement heatmap [8].

Table 13
The results of two-sided Wilcoxon Signed-Rank test with $\alpha = 0.05$ for (F1–F7), with 30 dimensions.

F		EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F1	*p*-value	1	1	1	1.7235e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1	1.7344e−06	1	1
	(T+,T−)	(0,0)	(0,0)	(0,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(0,0)	(465,0)	(0,0)	(0,0)
	winner	=	=	=	+	+	+	+	+	+	+	+	+	=	+	=	=
F2	*p*-value	3.4142e−07	1	1	1.56e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1	1
	(T+,T−)	(351,0)	(0,0)	(0,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(0,0)	(0,0)
	winner	+	=	=	+	+	+	+	+	+	+	+	+	+	+	=	=
F3	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	1.7289e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1	1
	(T+,T−)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(0,0)	(0,0)
	winner	+	+	+	+	+	+	+	+	+	+	+	+	+	+	=	=
F4	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.5	1
	(T+,T−)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(3,0)	(0,0)
	winner	+	+	+	+	+	+	+	+	+	+	+	+	+	+	=	=
F5	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	0.0087297	0.38203	1.7344e−06	1.7344e−06	0.82901	0.057096	0.047162	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	5.792e−05
	(T+,T−)	(0,465)	(0,465)	(0,465)	(360,105)	(275,190)	(0,465)	(0,465)	(222,243)	(140,325)	(329,136)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(37,428)
	winner	−	−	−	+	=	−	−	=	=	+	−	−	−	−	−	−
F6	*p*-value	1	1	1	0.0625	1	1	1	1.6699e−06	1.7344e−06	1.7213e−06	1.6678e−06	1.73e−06	2.5288e−06	7.9428e−06	1	1
	(T+,T−)	(0,0)	(0,0)	(0,0)	(15,0)	(0,0)	(0,0)	(0,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(435,0)	(351,0)	(0,0)	(0,0)
	winner	=	=	=	=	=	=	=	+	+	+	+	+	+	+	=	=
F7	*p*-value	0.007271	0.0036094	0.00096266	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.00077122	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.00083071	0.051931
	(T+,T−)	(363,102)	(374,91)	(393,72)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(396,69)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(70,395)	(138,327)
	winner	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−	=
Total	(+/ = /−)	**4/2/1**	**3/3/1**	**3/3/1**	**6/1/0**	**5/2/0**	**5/1/1**	**5/1/1**	**6/1/0**	**6/1/0**	**7/0/0**	**6/0/1**	**6/0/1**	**5/1/1**	**6/0/1**	**0/5/2**	**0/6/1**

Table 14
The results of two-sided Wilcoxon signed-rank test with $\alpha = 0.05$ for (F8–F13), with 30 dimensions.

F		EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F8	*p*-value	0.40483	1.7344e−06	1.7344e−06	0.10201	1.7344e−06	1.7344e−06	1.7344e−06	1.6394e−05	1.7344e−06	3.8822e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7333e−06	0.00028308	1.921e−06
	(T+,T−)	(192,273)	(465,0)	(0,465)	(312,153)	(465,0)	(0,465)	(465,0)	(442,23)	(0,465)	(8,457)	(0,465)	(0,465)	(0,465)	(0,465)	(409,56)	(1,464)
	winner	=	+	−	=	+	−	+	+	−	−	−	−	−	−	+	−
F9	*p*-value	1	1	1	1.7344e−06	1.7268e−06	1.7344e−06	1.7224e−06	1.7344e−06	1.7344e−06	1.7344e−06	2.3018e−06	0.00024414	1.7257e−06	6.0889e−06	1	1
	(T+,T−)	(0,0)	(0,0)	(1,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(351,0)	(91,0)	(465,0)	(325,0)	(0,0)	(0,0)
	winner	=	=	=	+	+	+	+	+	+	+	+	+	+	+	=	=
F10	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	4.7292e−06	1.7344e−06	0.0027653	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.9209e−06	3.1817e−06	1.7344e−06	1.734e−06
	(T+,T−)	(0,465)	(0,465)	(0,465)	(455,10)	(0,465)	(87,378)	(465,0)	(465,0)	(0,465)	(465,0)	(465,0)	(465,0)	(464,1)	(459,6)	(0,465)	(0,465)
	winner	−	−	−	+	−	−	+	+	−	+	+	+	+	+	−	−
F11	*p*-value	1	1	1	4.0053e−05	1.8037e−05	1	0.03125	1.7344e−06	1.7344e−06	1.7344e−06	8.2081e−06	1.7344e−06	1.229e−05	1.7579e−05	3.7896e−06	1
	(T+,T−)	(0,0)	(1,0)	(0,0)	(253,0)	(300,0)	(0,0)	(21,0)	(465,0)	(465,0)	(465,0)	(351,0)	(465,0)	(325,0)	(300,0)	(406,0)	(0,0)
	winner	=	=	=	+	+	=	+	+	+	+	+	+	+	+	+	=
F12	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	0.020671	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.0087297	0.020671	0.020671	0.38203	1.7344e−06	1.734e−06
	(T+,T−)	(0,465)	(465,0)	(465,0)	(345,120)	(0,465)	(0,465)	(0,465)	(465,0)	(465,0)	(465,0)	(360,105)	(345,120)	(345,120)	(275,190)	(465,0)	(465,0)
	winner	−	+	+	+	−	−	−	+	+	+	+	+	+	=	+	+
F13	*p*-value	0.011748	1.7344e−06	0.0003065	0.0036094	2.5967e−05	1.7344e−06	1.7344e−06	1.4936e−05	0.00035888	1.9209e−06	0.44052	4.2857e−06	0.047162	0.075213	1.7344e−06	1.734e−06
	(T+,T−)	(355,110)	(465,0)	(408,57)	(374,91)	(28,437)	(0,465)	(0,465)	(443,22)	(406,59)	(464,1)	(270,195)	(456,9)	(329,136)	(319,146)	(465,0)	(465,0)
	winner	+	+	+	+	−	−	−	+	+	+	=	+	+	=	+	+
Total	(+/ = /−)	1/3/2	3/2/1	2/2/2	5/1/0	3/0/3	1/1/4	4/0/2	6/0/0	4/0/2	5/0/1	4/1/1	5/0/1	5/0/1	3/2/1	4/1/1	2/2/2

Table 15
The results of two-sided Wilcoxon signed-rank test with $\alpha = 0.05$ for (F14–F23).

F		EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F14	*p*-value	5.1582e−06	0.0011138	0.76552	0.82781	0.0052197	2.1148e−06	4.2832e−06	0.0047739	0.7139	0.29082	2.1148e−06	2.1148e−06	4.6728e−05	2.1148e−06	2.3534e−06	0.10201
	(T+,T−)	(0,351)	(391,74)	(218,247)	(180,198)	(285,66)	(0,406)	(456,9)	(15,121)	(126,105)	(169,266)	(0,406)	(0,406)	(28,378)	(0,406)	(462,3)	(312,153)
	winner	−	+	=	=	+	−	+	−	=	=	−	−	−	−	+	=
F15	*p*-value	0.057096	0.45281	0.7971	6.3391e−06	2.3704e−05	1.7344e−06	1.7344e−06	7.6909e−06	0.55774	0.00020515	0.014795	0.0027653	1.7344e−06	3.1123e−05	0.0077309	2.5967e−05
	(T+,T−)	(140,325)	(196,269)	(220,245)	(452,13)	(438,27)	(0,465)	(465,0)	(450,15)	(261,204)	(52,413)	(114,351)	(87,378)	(0,465)	(30,435)	(362,103)	(28,437)
	winner	=	=	=	+	+	−	+	+	=	−	−	−	−	−	+	−
F16	*p*-value	1.7289e−06	1.7344e−06	2.3534e−06	1.7289e−06	1.73e−06	1.7289e−06	1.73e−06	7.3129e−06	1.73e−06	1.7289e−06	1.7289e−06	1.7289e−06	1.7289e−06	1.7289e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(0,465)	(465,0)	(462,3)	(0,465)	(0,465)	(0,465)	(0,465)	(14.5,450.5)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(465,0)	(465,0)
	winner	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+	+
F17	*p*-value	1.7311e−06	1.7344e−06	1.7344e−06	1.7311e−06	1.7311e−06	1.7311e−06	1.7311e−06	3.0229e−06	1.7311e−06	1.7311e−06	1.7311e−06	1.7311e−06	1.7311e−06	1.7311e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(0,465)	(465,0)	(465,0)	(0,465)	(0,465)	(0,465)	(0,465)	(5.5,459.5)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(465,0)	(465,0)
	winner	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+	+
F18	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	2.3534e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(0,465)	(465,0)	(465,0)	(0,465)	(0,465)	(0,465)	(0,465)	(3,462)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(465,0)	(465,0)
	winner	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+	+
F19	*p*-value	3.0869e−05	1.7344e−06	1.7344e−06	1.7159e−06	1.7159e−06	1.7159e−06	1.7159e−06	0.0019494	1.7159e−06	1.7159e−06	1.7159e−06	1.7159e−06	1.7159e−06	1.7159e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(30,435)	(465,0)	(465,0)	(0,465)	(0,465)	(0,465)	(0,465)	(82,383)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(0,465)	(465,0)	(465,0)
	winner	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+	+
F20	*p*-value	0.064261	1.4936e−05	1.4936e−05	0.0011492	0.62947	0.76193	1.6919e−07	1.7344e−06	0.24349	0.16028	0.36858	0.97329	0.97452	0.0071618	1.7344e−06	1.7344e−06
	(T+,T−)	(301,134)	(443,22)	(443,22)	(365,70)	(196,239)	(204,231)	(0,435)	(465,0)	(270,165)	(299,166)	(275,190)	(216,219)	(234,231)	(110,355)	(465,0)	(465,0)
	winner	=	+	+	+	=	=	−	+	=	=	=	=	=	−	+	+
F21	*p*-value	0.40388	0.0064242	0.18462	0.14959	0.05729	1.9138e−05	0.0024182	0.11093	0.00039883	0.10167	0.95638	0.23922	0.077887	0.00014896	2.3534e−06	1.7344e−06
	(T+,T−)	(132,193)	(365,100)	(297,168)	(302,163)	(286,120)	(29,406)	(315,63)	(310,155)	(403.5,61.5)	(144,291)	(215,220)	(164,271)	(138,297)	(45,361)	(462,3)	(465,0)
	winner	=	+	=	=	=	−	+	=	+	=	=	=	=	−	+	+
F22	*p*-value	0.059012	0.0003065	1.9729e−05	0.36714	0.00012349	1.0144e−07	2.1923e−06	0.00026134	0.014706	0.0016369	0.0016369	0.0001254	0.0010045	7.5865e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(125,281)	(408,57)	(440,25)	(275,190)	(54,381)	(0,465)	(0,300)	(410,55)	(329,106)	(82,353)	(82,353)	(56,379)	(84,381)	(29,406)	(465,0)	(465,0)
	winner	=	+	+	=	−	−	−	+	+	−	−	−	−	−	+	+
F23	*p*-value	5.8161e−06	2.3704e−05	1.7344e−06	0.095815	3.5327e−06	6.7988e−08	1.3944e−05	2.5967e−05	0.019065	8.6828e−05	0.0011339	3.5327e−06	3.5327e−06	3.5327e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(29,406)	(438,27)	(465,0)	(312,153)	(30,435)	(0,465)	(26,352)	(437,28)	(345,120)	(59,406)	(87,378)	(30,435)	(30,435)	(30,435)	(465,0)	(465,0)
	winner	−	+	+	=	−	−	−	+	+	−	−	−	−	−	+	+
Total	(+/=/−)	0/4/6	9/1/0	7/3/0	2/4/4	2/2/6	0/1/9	3/0/7	4/1/5	3/3/4	0/3/7	0/2/8	0/2/8	0/2/8	0/0/10	10/0/0	8/1/1

Table 16
The results of two-sided Wilcoxon Signed-Rank test with $\alpha = 0.05$ for (F24–F29).

F		EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F24	*p*-value	0.047144	1.7344e−06	1.7344e−06	1.3565e−05	0.10182	0.0027646	1.7344e−06	1.7344e−06	0.00025909	0.16502	0.382	0.67327	0.97539	0.014793	1.7344e−06	1.7344e−06
	(T+,T−)	(329,136)	(465,0)	(465,0)	(444,21)	(312,153)	(87,378)	(0,465)	(465,0)	(410,55)	(165,300)	(275,190)	(212,253)	(234,231)	(114,351)	(465,0)	(465,0)
	winner	+	+	+	+	=	−	−	+	+	=	=	=	=	−	+	+
F25	*p*-value	3.1817e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.99179	1.7344e−06	1.7344e−06	1.7344e−06	3.1817e−06	2.3704e−05	1.0246e−05	5.7924e−05	0.0038542	1.7344e−06	1.7344e−06
	(T+,T−)	(459,6)	(465,0)	(465,0)	(465,0)	(465,0)	(232,233)	(465,0)	(465,0)	(465,0)	(459,6)	(438,27)	(447,18)	(428,37)	(373,92)	(465,0)	(465,0)
	winner	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+
F26	*p*-value	0.0038542	0.00022248	1.7344e−06	2.3534e−06	0.0043896	0.70356	0.94261	0.0001057	2.1266e−06	0.0031618	0.95899	0.071903	0.10201	0.32857	1.7344e−06	1.7344e−06
	(T+,T−)	(373,92)	(412,53)	(465,0)	(462,3)	(371,94)	(251,214)	(229,236)	(421,44)	(463,2)	(376,89)	(230,235)	(320,145)	(312,153)	(185,280)	(465,0)	(465,0)
	winner	+	+	+	+	+	=	=	+	+	+	=	=	=	=	+	+
F27	*p*-value	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.9209e−06	1.7344e−06	4.0715e−05	2.3534e−06	1.7344e−06	0.00014773	3.4053e−05	1.2381e−05	8.4661e−06	1.3601e−05	1.7344e−06	1.7344e−06
	(T+,T−)	(465,0)	(465,0)	(465,0)	(465,0)	(464,1)	(465,0)	(432,33)	(462,3)	(465,0)	(417,48)	(434,31)	(445,20)	(449,16)	(444,21)	(465,0)	(465,0)
	winner	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F28	*p*-value	4.7265e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.31849	3.1817e−06	2.3534e−06	1.7344e−06	7.6866e−06	3.5152e−06	5.2165e−06	5.307e−05	0.00013589	1.7344e−06	1.7344e−06
	(T+,T−)	(455,10)	(465,0)	(465,0)	(465,0)	(465,0)	(184,281)	(459,6)	(462,3)	(465,0)	(450,15)	(458,7)	(454,11)	(429,36)	(418,47)	(465,0)	(465,0)
	winner	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+
F29	*p*-value	2.8786e−06	1.7344e−06	1.7344e−06	1.9197e−06	4.2857e−06	0.38192	1.7344e−06	9.711e−05	1.7344e−06	1.6394e−05	3.8822e−06	2.3534e−06	6.3198e−05	0.00020515	1.7268e−06	2.352e−06
	(T+,T−)	(460,5)	(465,0)	(465,0)	(464,1)	(456,9)	(275,190)	(465,0)	(422,43)	(465,0)	(442,23)	(457,8)	(462,3)	(427,38)	(413,52)	(465,0)	(462,3)
	winner	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+	+
Total	(+/=/−)	6/0/0	6/0/0	6/0/0	6/0/0	5/1/0	1/4/1	4/1/1	6/0/0	6/0/0	5/1/0	4/2/0	4/2/0	4/2/0	4/1/1	6/0/0	6/0/0

Table 17
The results of two-sided Wilcoxon Signed-Rank test with $\alpha = 0.05$ for CEC2019 functions.

F		EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
F30	*p*-value	0.034897	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	9.5826e−05	0.16285	1.7344e−06	5.221e−06	0.028582	0.12206
	(T+,T−)	(315,120)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(27,351)	(282,153)	(465,0)	(0,378)	(98,280)	(289,146)
	winner	+	+	+	+	+	+	+	+	+	+	−	=	+	−	−	=
F31	*p*-value	3.1817e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	2.3534e−06	2.3534e−06	1.7344e−06	0.024308
	(T+,T−)	(459,6)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(462,3)	(462,3)	(465,0)	(123,342)
	winner	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−
F32	*p*-value	0.13056	0.00011499	3.1817e−06	1.7344e−06	0.38201	1.7344e−06	0.00026134	1.7344e−06	1.7344e−06	0.25361	0.45281	0.078647	0.8774	0.22102	1.7344e−06	1.7344e−06
	(T+,T−)	(159,306)	(420,45)	(459,6)	(465,0)	(275,190)	(465,0)	(410,55)	(465,0)	(465,0)	(177,288)	(269,196)	(318,147)	(240,225)	(292,173)	(465,0)	(465,0)
	winner	=	+	+	+	=	+	+	+	+	=	=	=	=	=	+	+
F33	*p*-value	0.017518	1.3601e−05	1.7344e−06	1.9209e−06	1.7344e−06	2.1266e−06	1.7333e−06	2.3534e−06	1.7344e−06	6.8923e−05	0.028486	0.0053197	0.0003065	0.0068359	1.7344e−06	1.7344e−06
	(T+,T−)	(348,117)	(444,21)	(465,0)	(464,1)	(465,0)	(463,2)	(465,0)	(462,3)	(465,0)	(426,39)	(126,339)	(97,368)	(57,408)	(101,364)	(465,0)	(465,0)
	winner	+	+	+	+	+	+	+	+	+	+	−	−	−	−	+	+
F34	*p*-value	0.0098421	1.7344e−06	1.7344e−06	1.7344e−06	5.2165e−06	1.9209e−06	1.9209e−06	1.7344e−06	1.7344e−06	9.711e−05	0.11093	0.0014839	0.036826	0.0011973	1.7344e−06	1.7344e−06
	(T+,T−)	(358,107)	(465,0)	(465,0)	(465,0)	(454,11)	(464,1)	(1,464)	(465,0)	(465,0)	(422,43)	(155,310)	(78,387)	(131,334)	(75,390)	(465,0)	(465,0)
	winner	+	+	+	+	+	+	−	+	+	+	=	−	−	−	+	+
F35	*p*-value	4.2857e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	5.307e−05	0.00026134	1.7344e−06	1.7344e−06	0.047162	0.74987	0.00016046	0.17791	0.37094	1.7344e−06	1.7344e−06
	(T+,T−)	(456,9)	(465,0)	(465,0)	(465,0)	(465,0)	(429,36)	(410,55)	(465,0)	(465,0)	(329,136)	(248,217)	(416,49)	(298,167)	(189,276)	(465,0)	(465,0)
	winner	+	+	+	+	+	+	+	+	+	+	=	+	=	=	+	+
F36	*p*-value	0.0018326	3.5152e−06	1.7344e−06	1.9209e−06	1.9209e−06	0.32857	1.7344e−06	2.3534e−06	2.3534e−06	0.00011499	0.0082167	0.036826	0.010444	0.0021053	1.7344e−06	1.7344e−06
	(T+,T−)	(384,81)	(458,7)	(465,0)	(464,1)	(464,1)	(185,280)	(465,0)	(462,3)	(462,3)	(420,45)	(104,361)	(131,334)	(108,357)	(83,382)	(465,0)	(465,0)
	winner	+	+	+	+	+	=	+	+	+	+	−	−	−	−	+	+
F37	*p*-value	0.11561	0.68836	2.6033e−06	2.3534e−06	9.3157e−06	0.0024147	1.7344e−06	0.0029575	2.3534e−06	0.027029	0.00013595	5.7517e−06	2.3704e−05	5.7924e−05	1.7344e−06	2.6033e−06
	(T+,T−)	(156,309)	(252,213)	(461,4)	(462,3)	(448,17)	(380,85)	(465,0)	(377,88)	(462,3)	(340,125)	(47,418)	(12,453)	(27,438)	(37,428)	(465,0)	(461,4)
	winner	=	=	+	+	+	+	+	+	+	+	−	−	−	−	+	+
F38	*p*-value	0.0043896	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	1.7344e−06	0.057096	1.7344e−06	1.9209e−06	7.6909e−06	0.00024118	0.031603	0.0077309	8.4661e−06	1.7344e−06	1.7344e−06
	(T+,T−)	(371,94)	(465,0)	(465,0)	(465,0)	(465,0)	(465,0)	(140,325)	(465,0)	(464,1)	(450,15)	(411,54)	(337,128)	(362,103)	(449,16)	(465,0)	(465,0)
	winner	+	+	+	+	+	+	=	+	+	+	+	+	+	+	+	+
F39	*p*-value	0.0011973	1.7344e−06	1.7344e−06	1.7344e−06	2.3704e−05	1.7344e−06	0.31849	1.2381e−05	0.00038811	2.6033e−06	0.67328	0.84508	0.035009	0.10639	1.2381e−05	0.13591
	(T+,T−)	(390,75)	(465,0)	(465,0)	(465,0)	(438,27)	(465,0)	(184,281)	(445,20)	(405,60)	(461,4)	(253,212)	(242,223)	(130,335)	(311,154)	(445,20)	(305,160)
	winner	+	+	+	+	+	+	=	+	+	+	=	=	−	=	+	=
Total	(+/ = /−)	8/2/0	9/1/0	10/0/0	10/0/0	9/1/0	9/1/0	7/2/1	10/0/0	10/0/0	9/1/0	2/4/4	3/3/4	3/2/5	2/3/5	9/0/1	7/2/1

Table 18
Results based on test category of two-sided Wilcoxon Signed-Rank test ($\alpha = 0.05$).

Test category	EO	GWO	WOA	MFO	PSO	DE	GSA	SSA	SHO	GA-MPC	LSHADE	LSHADE -cnEpSin	LSHADE -SPACMA	SASS	AOA	HGSO
Unimodals	4/2/1	3/3/1	3/3/1	6/1/0	5/2/0	5/1/1	5/1/1	6/1/0	6/1/0	7/0/0	6/0/1	6/0/1	5/1/1	6/0/1	0/5/2	0/6/1
Multimodals	1/3/2	3/2/1	2/2/2	5/1/0	3/0/3	1/1/4	4/0/2	6/0/0	4/0/2	5/0/1	4/1/1	5/0/1	5/0/1	3/2/1	4/1/1	2/2/2
Fixed-dimension multimodal	0/4/6	9/1/0	7/3/0	2/4/4	2/2/6	0/1/9	3/0/7	4/1/5	3/3/4	0/3/7	0/2/8	0/2/8	0/2/8	0/0/10	10/0/0	8/1/1
Composite	6/0/0	6/0/0	6/0/0	6/0/0	5/1/0	1/4/1	4/1/1	6/0/0	6/0/0	5/1/0	4/2/0	4/2/0	4/2/0	4/1/1	6/0/0	6/0/0
CEC2019	8/2/0	9/1/0	10/0/0	10/0/0	9/1/0	9/1/0	7/2/1	10/0/0	10/0/0	9/1/0	2/4/4	3/3/4	3/2/5	2/3/5	9/0/1	7/2/1
Total(+/ = /−)	**19/11/9**	**30/7/2**	**28/8/3**	**29/6/4**	**24/6/9**	**16/8/15**	**23/4/12**	**32/2/5**	**29/4/6**	**26/5/8**	**16/9/14**	**18/7/14**	**17/7/15**	**15/6/18**	**29/6/4**	**23/11/5**

Table 19
Ranking of algorithm according to Friedman ranking based on mean error value. (FR: Friedman Ranking).

S.N.	Algorithm	FR	Rank	S.N.	Algorithm	FR	Rank
1	ICO	5.1154	1	10	SHO	11.641	15
2	EO	7.0897	6	11	GA_MPC	9.6667	10
3	GWO	9.6923	11	12	LSHADE	7.3974	7
4	WOA	10.8718	14	13	LSHADE_cnEpSin	7	5
5	MFO	13.0513	17	14	LSHADE_SPACMA	6.5769	4
6	PSO	9.641	9	15	SASS	5.5	2
7	DE	6.5	3	16	AOA	12.9744	16
8	GSA	9.0641	8	17	HGSO	10.6026	12
9	SSA	10.6154	13				

Table 20
Comparison of ICO optimization results with literature for the pressure vessel design problem.

Algorithm	Optimal Value				Optimal Cost
	T_s	T_h	R	L	
ICO	0.78168	0.38639	40.5017	197.4812	5891.383
GSA [13]	1.125	0.625	55.98866	84.4542	8538.836
MSCA [26]	0.779256	0.3996	40.32545	199.9213	5935.716
EO [1]	0.8125	0.4375	42.09844	176.63659	6059.7143
PRO [25]	0.7445	0.4424	38.48998	200	6050.713
SMO [24]	0.8242	0.4072	42.36585	173.7973	6021.629
WOA [13]	0.8125	0.4375	42.09827	176.6389	6059.741
GWO [8]	0.8125	0.4345	42.08918	176.7587	6051.564
PSO [74]	0.8125	0.4375	42.09127	176.7465	6061.078
GA [35]	0.8125	0.4345	40.3239	200	6288.745

Table 21
Comparison of ICO statistical results with literature for the pressure vessel design problem.

Algorithm	Min	mean	max	Std	Function evaluation
ICO	5891.383	6147.1439	6814.5663	235.9747	40000
GSA [13]	8538.836	8932.95	65535	683.5475	7110
EO [1]	6059.714	6668.114	7544.493	566.24	15000
WOA [13]	6059.741	6068.05	65535	65.6519	6300
PSO [74]	6061.078	6531.1	65535	154.3716	14790
CPSO [74]	6061.078	6147.133	6363.804	86.4545	200000
HGA(1)) [75]	6065.821	6632.376	8248.003	515	80000
HGA(2) [75]	6832.584	7187.314	8012.615	276	80000
T-Cell [76]	6390.554	6737.065	7694.066	357	80000
DTS-GA [77]	6059.946	6177.253	6469.322	130.92	80000
HAIS-GA [66]	6061.123	6743.085	7368.06	457.99	150000
BFOA [78]	6060.46	6074.625	65535	156	48000
ES [79]	6059.746	6850	7332.87	426	25000

4.3. Tension/compression spring design

This problem includes minimizing the weight of a tension/compression spring to impose constraints on the minimum deviation, stretch tension, and boiling frequency as well as limitations on external thickness and design variables in Fig. 13. The design variables included the core average diameter (D), wire diameter (d) and the number of active cores. The problem can be defined as follows:

Minimization of

$$f(x) = (N+2)\,Dd^2$$

For

$$g_1(X) = 1 - \frac{D^3N}{71785d^4} \le 0$$
$$g_2(X) = \frac{4D^2 - dD}{12566\left(Dd^3 - d^4\right)} + \frac{1}{5108d^2} - 1 \le 0 \quad (27)$$
$$g_3(X) = 1 - \frac{140.45d}{D^2N}$$
$$g_4(X) = \frac{D+d}{1.5} - 1 \le 0$$

Table 22
Comparison of ICO optimization results with literature for the welded beam design problem.

Algorithm	Optimal Value				Optimal Cost
	H	L	t	B	
ICO	0.204	7.1773	9.0367	0.20573	2.2241
GSA [13]	0.182129	3.856979	10	0.202376	1.879952
MSCA [26]	0.20545	3.2524	9.0576	0.20568	1.6979
WOA [13]	0.205396	3.484293	9.037426	0.206276	1.730499
EO [1]	0.2057	3.4705	9.03664	0.2057	1.7249
GWO [8]	0.205676	3.478377	9.03681	0.205778	1.72624
GA [35]	65535	65535	65535	65535	1.8245
HS [80]	0.2442	6.2231	8.2915	0.2433	2.3807
Random [81]	0.4575	4.7313	5.0853	0.66	4.1185
Simple [81]	0.2792	5.6256	7.7512	0.2796	2.5307
David [81]	0.2434	6.2552	8.2915	0.2444	2.3841
Approx [81]	0.2444	6.2189	8.2915	0.2444	2.3815

The following ranges were used for variables:

$$0.05 \le x_1 \le 2.0,\ 0.25 \le x_2 \le 1.3,\ 2.0 \le x_3 \le 15.0$$

This mathematical modeling problem can be solved by using mathematical techniques such as constraint correction in a cost function and penalty functions [89] or by using metaheuristic algorithms. In [26], Chen et al. adopted MSCA, the modified version of SCA, to solve this problem and obtained an optimal weight of

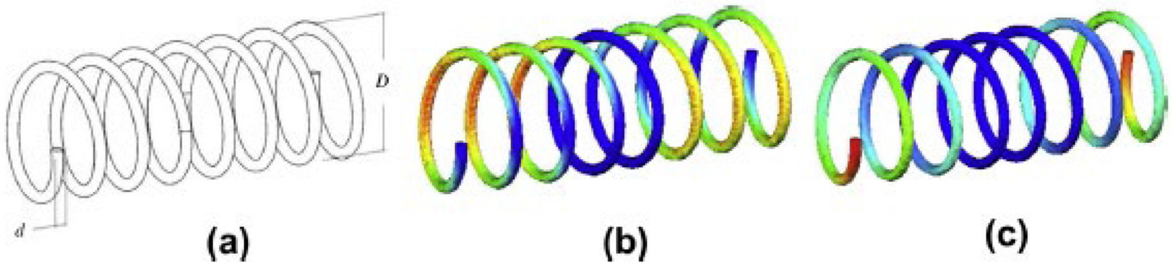

Fig. 13. Tension/compression spring: (a) schematic, (b) stress heatmap (c) displacement heatmap [8].

Table 23
Comparison of ICO statistical results with literature for the welded beam design problem.

Algorithm	Min	mean	Max	Std	Function evaluation
ICO	2.2241	2.485	3.0019	0.23121	40000
GSA [13]	1.879952	3.5761	65535	0.2874	10750
EO [1]	1.724853	1.726482	1.736725	0.003257	15000
WOA [13]	1.730499	1.732	65535	0.0226	9900
PSO [74]	65535	1.7422	65535	1.01275	13770
SBM [82]	2.4426	2.5215	2.6315	65535	19259
BFOA [78]	2.3868	2.404	65535	0.016	48000
SCA [83]	2.3854	3.2551	6.3996	0.959	33095
EA [84]	2.3816	65535	2.38297	0.00034	28897
T-Cell [76]	2.3811	2.4398	2.7104	0.09314	320000
FSA [85]	2.3811	2.4041	2.4889	65535	56243
IPSO [86]	2.381	2.3819	65535	0.00523	30000
HSA-GA [87]	2.25	2.26	2.28	0.0078	26466
CDE [88]	1.7335	1.768158	1.824105	0.022194	240000
CPSO [74]	1.728	1.748831	1.782143	0.012926	200000

Table 24
Comparison of ICO optimization results with literature for tension compression spring design problem.

Algorithm	Optimal value			Optimal cost
	D	N	D	
ICO	0.051686	0.35663	11.2939	0.012665
MSCA [26]	0.050276	0.32368	13.52541	0.012702
WOA [13]	0.051207	0.345215	12.00403	0.012676
EO [1]	0.05162	0.355054	11.38797	0.012666
GWO [8]	0.05169	0.356737	11.28885	0.012666
GSA [13]	0.051668	0.356199	11.3207	0.012667
PSO [74]	0.051728	0.357644	11.24454	0.012674
GA [35]	0.05148	0.351661	11.6322	0.012704
RO [51]	0.05137	0.349096	11.76279	0.012679
DE [88]	0.051609	0.354714	11.41083	0.012670

0.0126670. Coello et al. [35]used the GA to solve the problem and obtained an optimal weight of 0.0127048.

The WOA, EO, GWO, PSO, GA, RO, HIS, DE and constraint correction algorithms were utilized to solve this problem. According to Table 24, ICO (with an appropriate weight of 0.126652331224549) proved to be the best methods among the standard metaheuristic algorithms and mathematical techniques used in this paper. This finding indicates that the ICO acted as an appropriate tool in this problem. Table 25 shows the minimum, maximum, mean, and standard deviation calculated for this algorithm in comparison with other methods. Above mentioned values were available for comparison.

5. Discussion

The proposed method is compared with 16 different methods including SHO, SSA, GSA, DE, PSO, MFO, WOA, GWO, EO, LSHADE, LSHADE-cnEpSin, LSHADE-SPACMA, SASS, GA-MPC, AOA and HGSO. Evaluation results stated completely in Section 3 are presented in this part in summary.

The results of ICO comparison with existing methods in unimodal functions are shown in 8, and Wilcoxon Signed-Rank Test results are demonstrated in Table 13. According to the results obtained in this category of functions, ICO has the better or equal performance than other methods except F5 and F7. In the function F7, ICO is superior to all methods except HGSO. These evaluations show the higher power of the proposed algorithm in terms of exploitation.

The statistical results of algorithms in multimodal and fixed-dimension multimodal are respectively presented in Tables 9 and 10, and the results of Wilcoxon Signed-Rank Test are respectively displayed in Tables 14 and 15. According to obtained results, DE algorithm has the better performance than all methods in this class of functions. The performance success of ICO in fixed-dimension multimodal functions is relatively low compared to others. This fits with "No-Free-Lunch" (NFL) theorem.

The results of ICO comparison with other methods in composite functions are presented in Table 11. According to the mean of obtained results, ICO outperformed other methods in this class of functions. Also the results of Wilcoxon Signed-Rank Test presented in Table 16 shows that ICO and DE have the similar results in this class of functions, and ICO is superior to other methods. According to these evaluations, it can be said that ICO has the higher power to solve the complex problems and to create a balance between exploitation and exploration.

Tables 12 and 17 respectively show evaluation results and Wilcoxon Signed-Rank Test results of ICO method in comparison to other methods in CEC2019 functions. According to Wilcoxon Signed-Rank Test results, ICO is superior to 12 methods including EO, GWO, WOA, MFO, PSO, DE, GSA, SSA, SHO, GA-MPC, AOA, and HGSO. Also according to the mean of obtained results, comparison of the algorithms is presented in Table 19 by considering 39 different functions on the basis of Friedman test. According to the obtained results, ICO has the highest rank than other algorithms. In Fig. 4, trajectory in the 1st dimension, search history, the mean of solutions fitness and convergence diagrams are shown in Fig. 4. Figs. 5–7 present the convergence diagram of the proposed method in comparison with different algorithms in 39 benchmark functions. According to diagram of search history, it can be concluded that ICO effectively creates a balance between exploitation and exploration so that solutions move toward the optimum point. According to the dimension of trajectory, it can be said that ICO converges to a point in search space. In order to confirm that this behavior results in improving the solutions fitness, the mean of solutions fitness and ICO convergence diagram are displayed. Diagrams clearly show descending behavior in all test functions. It proves that ICO considerably improves approximated optimum precision during the iterations.

The scalability of the proposed method in comparison to other well-known methods in EO, GWO and PSO with different dimensions of 100–1000 are demonstrated in Fig. 8. Also ICO convergence diagram is shown in Fig. 9 in comparison to other methods

Table 25
Comparison of ICO statistical results with literature for the tension compression spring design problem.

Algorithm	Min	Mean	Max	Std	Function evaluation
ICO	0.012665	0.013653	0.016282	0.0011985	30000
GSA [13]	0.012702	0.0136	65535	0.002630	4980
WOA [13]	0.012676	0.0127	65535	0.0003	4410
PSO [74]	0.012675	139	65535	0.0033	5460
SI [90]	0.01306	0.015526	0.018992	65535	20000
GA(1) [35]	0.012704	0.012769	0.012822	3.93E−05	65535
T-Cell [76]	0.012665	0.013309	0.012732	0.000094	36000
GA(2) [91]	0.012681	0.012742	0.012973	0.000095	80000
SCA [83]	0.012669	0.012922	0.016717	0.000592	25167
EO [1]	0.012666	0. 01302	0.013997	0.000391	15000
CDE [88]	0.01267	0.012703	0.01279	2.07E−05	240000

with low and high and very high dimensions. According to the obtained results, by increasing dimensions, the performance of PSO, GWO and EO algorithms decreases, but in ICO method, by increasing dimensions, trivial reduction is observed in its precision. It shows the higher scalability of the proposed method in comparison to available methods.

The sum of execution time in various algorithms is shown in Fig. 10. According to the figure, execution time of ICO is less than execution time of well-known and efficient algorithms like LSHADE-cnEpSin, LSHADE-SPACMA and DE.

In Tables 20–25, the results obtained by different algorithms to solve three engineering problems are stated. The proposed method could yield the best results in two engineering problems.

The logic behind the performance power of the proposed technique is to use the operators to increase the amount of exploitation and exploration and to maintain population diversity. The amount of exploitation increases by clonal operator used in ICO. Cloning each parent is performed by considering its fitness. The cloning is performed in two ways. The local search is increased by cloning near the parent. The solutions are localized to search the promising areas by cloning each solution in trajectory of temporary target. In this way, exploitation is increased intelligently by clonal operator. Also exploration is increased by initialization operator through the chaos theory. In addition, the proposed conservative selection operator is effective in increasing exploration by maintaining diversity. By the proposed selected operator, both the elited solutions localized locally and in trajectory of temporary target, along with elited parents are selected. In order to increase population diversity, when most solutions are close together, random members are added to the population. Therefore, by maintaining population diversity, due to this operator, it helps the solutions selected to transfer next generation cannot be localized in an area. A balance is created between exploitation and exploration by using these operators. The proposed method can have the higher efficiency to solve optimization problems by escaping from local optimization.

6. Conclusion

This paper proposed an evolutionary algorithm with the novel clonal and conservative selection operators to solve optimization problems. Thirty-nine test functions consisting of five different types including unimodal, multimodal, fixed-dimensional multimodal, composite and CEC 2019 were employed in order to test the performance of proposed optimizer in respect to exploration, exploitation, local optima avoidance and convergence.

For efficiency analysis, the proposed method was compared with EO, SASS, GWO, WOA, SSA, MFO, GSA, SHO, AOA and HGSO introduced as recently developed algorithms. It was also compared with DE and PSO, known as the most well-known algorithms, as well as GA–MPC, LSHADE, LSHADE–cnEpSin and LSHADE–SPACMA selected as the best performer of the IEEE CEC competitions. The obtained results of ICO are compared with the results of other optimization algorithms using statistical analysis, scalability analysis, Wilcoxon Signed-Rank Test, Friedman test, computational time analysis and convergence analysis. According to the results, the proposed method was introduced as the best technique ranked through the Friedman test in 39 benchmark functions. This method was also used to solve three well-known engineering problems of pressure vessel design, welded beam design and weight minimization of an item. The efficiency of this method was also evaluated in terms of solving these problems. The proposed method yielded the best results in two of these engineering problems.

ICO algorithm can reach the result by less iteration, while the number of function fitness evaluation increases on the basis of number of clones, and this is the weakness of proposed method. Computing the function fitness can be reduced in each iteration considerably by presenting new solutions in the future. Another disadvantage of the proposed method is the presence of many variables. The algorithm can be improved by removing or setting these parameters automatically in the future. Also execution time of the proposed method is less than well-known and efficient methods, while its computation time can be reduced by improving the algorithm in the future. Also, it is recommended to employ the ICO to solve binary and multi-objective optimization problems in future studies.

CRediT authorship contribution statement

Vahideh Sahargahi: Conceptualization, Writing - original draft, Software. **Vahid Majidnezhad:** Conceptualization, Supervision, Project administration, Validation. **Saeid Taghavi Afshord:** Writing - review & editing, Validation. **Yasser Jafari:** Writing - review & editing, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190, http://dx.doi.org/10.1016/j.knosys.2019.105190.

[2] A. Kumar, R.K. Misra, D. Singh, S. Mishra, S. Das, The spherical search algorithm for bound-constrained global optimization problems, Appl. Soft Comput. 85 (2019) 105734, http://dx.doi.org/10.1016/j.asoc.2019.105734.

[3] X.-S. Yang, Firefly. algorithm, Stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84, http://dx.doi.org/10.1504/ijbic.2010.032124.

[4] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315, http://dx.doi.org/10.1016/j.cad.2010.12.015.

[5] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845, http://dx.doi.org/10.1016/j.cnsns.2012.05.010.

[6] A. Kaveh, N. Farhoudi, A new optimization method: Dolphin echolocation, Adv. Eng. Softw. 59 (2013) 53–70, http://dx.doi.org/10.1016/j.advengsoft.2013.03.004.

[7] A. Askarzadeh, Bird mating optimizer: An optimization algorithm inspired by bird mating strategies, Commun. Nonlinear Sci. Numer. Simul. 19 (4) (2014) 1213–1228, http://dx.doi.org/10.1016/j.cnsns.2013.08.027.

[8] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[9] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98, http://dx.doi.org/10.1016/j.advengsoft.2015.01.010.

[10] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249, http://dx.doi.org/10.1016/j.knosys.2015.07.006.

[11] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073, http://dx.doi.org/10.1007/s00521-015-1920-1.

[12] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513, http://dx.doi.org/10.1007/s00521-015-1870-7.

[13] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

[14] S. Mirjalili, S. Saremi, S.M. Mirjalili, L. d. S. Coelho, Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119, http://dx.doi.org/10.1016/j.eswa.2015.10.039.

[15] M. Yazdani, F. Jolai, Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm, J. Comput. Des. Eng. 3 (1) (2016) 24–36, http://dx.doi.org/10.1016/j.jcde.2015.06.003.

[16] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: A novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85, http://dx.doi.org/10.1016/j.compstruc.2016.01.008.

[17] S.H. Samareh Moosavi, V. Khatibi Bardsiri, Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation, Eng. Appl. Artif. Intell. 60 (2017) 1–15, http://dx.doi.org/10.1016/j.engappai.2017.01.006.

[18] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47, http://dx.doi.org/10.1016/j.advengsoft.2017.01.004.

[19] F. Fausto, E. Cuevas, A. Valdivia, A. González, A global optimization algorithm inspired in the behavior of selfish herds, Biosystems 160 (2017) 39–55, http://dx.doi.org/10.1016/j.biosystems.2017.07.010.

[20] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191, http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.

[21] M. Kumar, A.J. Kulkarni, S.C. Satapathy, Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology, Future Gener. Comput. Syst. 81 (2018) 252–272, http://dx.doi.org/10.1016/j.future.2017.10.052.

[22] S. Gupta, K. Deep, A novel random walk grey wolf optimizer, Swarm Evol. Comput. 44 (2019) 101–112, http://dx.doi.org/10.1016/j.swevo.2018.01.001.

[23] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris Hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872, http://dx.doi.org/10.1016/j.future.2019.02.028.

[24] S. Balochian, H. Baloochian, Social mimic optimization algorithm and engineering applications, Expert Syst. Appl. 134 (2019) 178–191, http://dx.doi.org/10.1016/j.eswa.2019.05.035.

[25] S.H. Samareh Moosavi, V.K. Bardsiri, Poor and rich optimization algorithm: A new human-based and multi populations algorithm, Eng. Appl. Artif. Intell. 86 (2019) 165–181, http://dx.doi.org/10.1016/j.engappai.2019.08.025.

[26] H. Chen, M. Wang, X. Zhao, A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems, Appl. Math. Comput. 369 (2020) 124872, http://dx.doi.org/10.1016/j.amc.2019.124872.

[27] R. Zhao, et al., Selfish herd optimizer with levy-flight distribution strategy for global optimization problem, Physica A 538 (2020) 122687, http://dx.doi.org/10.1016/j.physa.2019.122687.

[28] Y. Li, Y. Zhao, J. Liu, Dimension by dimension dynamic sine cosine algorithm for global optimization problems, Appl. Soft Comput. 98 (2021) 106933, http://dx.doi.org/10.1016/j.asoc.2020.106933.

[29] H. Chen, W. Li, X. Yang, A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems, Expert Syst. Appl. 158 (2020) 113612, http://dx.doi.org/10.1016/j.eswa.2020.113612.

[30] Y. Chen, D. Pi, An innovative flower pollination algorithm for continuous optimization problem, Appl. Math. Model. 83 (2020) 237–265, http://dx.doi.org/10.1016/j.apm.2020.02.023.

[31] J. Jiang, Y. Liu, Z. Zhao, TriTSA: Triple tree-seed algorithm for dimensional continuous optimization and constrained engineering problems, Eng. Appl. Artif. Intell. 104 (2021) 104303, http://dx.doi.org/10.1016/j.engappai.2021.104303.

[32] A. Mortazavi, Bayesian interactive search algorithm: A new probabilistic swarm intelligence tested on mathematical and structural optimization problems, Adv. Eng. Softw. 155 (2021) 102994, http://dx.doi.org/10.1016/j.advengsoft.2021.102994.

[33] Z. Cheng, H. Song, J. Wang, H. Zhang, T. Chang, M. Zhang, Hybrid firefly algorithm with grouping attraction for constrained optimization problem, Knowl.-Based Syst. 220 (2021) 106937, http://dx.doi.org/10.1016/j.knosys.2021.106937.

[34] Z. k. Feng, W. j. Niu, S. Liu, Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems, Appl. Soft Comput. 98 (2021) 106734, http://dx.doi.org/10.1016/j.asoc.2020.106734.

[35] C.A. Coello Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127, http://dx.doi.org/10.1016/S0166-3615(99)00046-9.

[36] J.R. Koza, Genetic Programming, MIT Press, 55 Hayward St. Cambridge, MA, United States, 1992.

[37] I. Rechenberg, Evolution strategy: Nature's way of optimization, in: H.W. Bergmann (Ed.), Optimization: Methods and Applications, Possibilities and Limitations, Berlin, Heidelberg, Springer Berlin Heidelberg, 1989, pp. 106–126.

[38] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (4) (2006) 355–366, http://dx.doi.org/10.1016/j.ecoinf.2006.07.003.

[39] K. Price, R. Storn, J.A. Lampinen, Differential Evolution, A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag Berlin Heidelberg, 2005.

[40] I.M. Ali, D. Essam, K. Kasmarik, A novel differential evolution mapping technique for generic combinatorial optimization problems, Appl. Soft Comput. 80 (2019) 297–309, http://dx.doi.org/10.1016/j.asoc.2019.04.017.

[41] L. De Castro, F. Von Zuben, Learning and optimization using the clonal selection principle, Evol. Comput., IEEE Trans. 6 (2002) 239–251, http://dx.doi.org/10.1109/TEVC.2002.1011539.

[42] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248, http://dx.doi.org/10.1016/j.ins.2009.03.004.

[43] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 27 Nov.-1 Dec. 1995, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[44] K. Chen, F. Zhou, Y. Wang, L. Yin, An ameliorated particle swarm optimizer for solving numerical optimization problems, Appl. Soft Comput. 73 (2018) 482–496, http://dx.doi.org/10.1016/j.asoc.2018.09.007.

[45] Q. Liu, L. Wu, W. Xiao, F. Wang, L. Zhang, A novel hybrid bat algorithm for solving continuous optimization problems, Appl. Soft Comput. 73 (2018) 67–82, http://dx.doi.org/10.1016/j.asoc.2018.08.012.

[46] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39, http://dx.doi.org/10.1109/MCI.2006.329691.

[47] X. Chu, et al., An artificial bee colony algorithm with adaptive heterogeneous competition for global optimization problems, Appl. Soft Comput. 93 (2020) 106391, http://dx.doi.org/10.1016/j.asoc.2020.106391.

[48] W. Al-Sorori, A.M. Mohsen, New caledonian crow learning algorithm: A new metaheuristic algorithm for solving continuous optimization problems, Appl. Soft Comput. 92 (2020) 106325, http://dx.doi.org/10.1016/j.asoc.2020.106325.

[49] F. Glover, M. Laguna, Tabu search, INFORMS J. Comput. 1 (3) (1989) 190–206.

[50] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, 25-28 Sept. 2007, 2007, pp. 4661–4667, http://dx.doi.org/10.1109/CEC.2007.4425083.

[51] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: A novel metaheuristic method, Comput. Struct. 139 (2014) 18–27, http://dx.doi.org/10.1016/j.compstruc.2014.04.005.

[52] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612, http://dx.doi.org/10.1016/j.asoc.2012.11.026.

[53] M.A. Eita, M.M. Fahmy, Group counseling optimization, Appl. Soft Comput. 22 (2014) 585–604, http://dx.doi.org/10.1016/j.asoc.2014.03.043.

[54] E. Bogar, S. Beyhan, Adolescent identity search algorithm (AISA): A novel metaheuristic approach for solving optimization problems, Appl. Soft Comput. 95 (2020) 106503, http://dx.doi.org/10.1016/j.asoc.2020.106503.

[55] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609, http://dx.doi.org/10.1016/j.cma.2020.113609.
[56] D. Połap, M. Woźniak, Red fox optimization algorithm, Expert Syst. Appl. 166 (2021) 114107, http://dx.doi.org/10.1016/j.eswa.2020.114107.
[57] M.S. Braik, Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Syst. Appl. 174 (2021) 114685, http://dx.doi.org/10.1016/j.eswa.2021.114685.
[58] H. Karami, M.V. Anaraki, S. Farzin, S. Mirjalili, Flow direction algorithm (FDA): A novel optimization approach for solving optimization problems, Comput. Ind. Eng. 156 (2021) 107224, http://dx.doi.org/10.1016/j.cie.2021.107224.
[59] M.E. Kivi, V. Majidnezhad, A novel swarm intelligence algorithm inspired by the grazing of sheep, J. Ambient Intell. Humaniz. Comput. (2021) http://dx.doi.org/10.1007/s12652-020-02809-y.
[60] E. Alba, B. Dorronsoro, The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, IEEE Trans. Evol. Comput. 9 (2) (2005) 126–142, http://dx.doi.org/10.1109/TEVC.2005.843751.
[61] X. Xia, S. Li, Research on Improved Chaotic Particle Optimization Algorithm Based on Complex Function, Frontiers in Physics, Brief Research Report, Vol. 8, (368) 2020, http://dx.doi.org/10.3389/fphy.2020.00368, (in English), 2020-September-09 2020.
[62] C. Fei, Z. Han, A novel chaotic optimization algorithm and its applications, J. Harbin Inst. Tech. (New Series) 17 (2010) 254–258.
[63] H. Lu, X. Wang, Z. Fei, M. Qiu, The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms, Math. Probl. Eng. 2014 (2014) 1–16.
[64] D. Tian, Particle swarm optimization with chaos-based initialization for numerical optimization, Intell. Autom. Soft Comput. 24 (2018) 331–342, http://dx.doi.org/10.1080/10798587.2017.1293881.
[65] Z. Xiaoming, T, Feng chaotic bean optimization algorithm, Soft Comput. 22 (2018) 67–77.
[66] C.A.C. Coello, N.C. Cortés, Hybridizing a genetic algorithm with an artificial immune system for global optimization, Eng. Optim. 36 (5) (2004) 607–634, http://dx.doi.org/10.1080/03052150410001704845.
[67] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667, http://dx.doi.org/10.1016/j.future.2019.07.015.
[68] S. Elsayed, R. Sarker, D. Essam, GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems, 2011, pp. 1034–1040.
[69] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation, CEC, 6-11 2014, 2014, pp. 1658–1665, http://dx.doi.org/10.1109/CEC.2014.6900380.
[70] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, 5-8 2017, 2017, pp. 372–379, http://dx.doi.org/10.1109/CEC.2017.7969336.
[71] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, Lshade with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, 5-8 2017, 2017, pp. 145–152, http://dx.doi.org/10.1109/CEC.2017.7969307.
[72] K. Price, N. Awad, M. Ali, P. Suganthan, Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization, Technical Report, Nanyang Technological University, 2018.
[73] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Inform. Sci. 176 (8) (2006) 937–971, http://dx.doi.org/10.1016/j.ins.2005.02.003.
[74] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99, http://dx.doi.org/10.1016/j.engappai.2006.03.003.
[75] H.S. Bernardino, H.J.C. Barbosa, A.C.C. Lemonge, L.G. Fonseca, A new hybrid AIS-GA for constrained optimization problems in mechanical engineering, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 1-6 2008, 2008, pp. 1455–1462, http://dx.doi.org/10.1109/CEC.2008.4630985.
[76] V.S. Aragón, S.C. Esquivel, C.A.C. Coello, A modified version of a T-cell algorithm for constrained optimization problems, Internat. J. Numer. Methods Engrg. 84 (3) (2010) 351–378, http://dx.doi.org/10.1002/nme.2904.
[77] C. Coello, Use of dominance-based tournament selection to handle constraints in genetic algorithms, Intell. Eng. Syst. Artif. Neural Netw. 11 (2001).
[78] E.M. Montes, B.H. Oca, Bacterial foraging for engineering design problems: preliminary results, in: 4th Mex. Congr. Evol. Comput., COMCEV'2008, Mexico, 2008, pp. 33–38.
[79] E. Mezura-Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (4) (2008) 443–473, http://dx.doi.org/10.1080/03081070701303470.
[80] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Engrg. 194 (36) (2005) 3902–3933, http://dx.doi.org/10.1016/j.cma.2004.09.007.
[81] K.M. Ragsdell, D.T. Phillips, Optimal design of a class of welded structures using geometric programming, J. Eng. Ind. 98 (3) (1976) 1021–1025, http://dx.doi.org/10.1115/1.3438995.
[82] S. Akhtar, K. Tai, T. Ray, A socio-behavioural simulation model for engineering design optimization, Eng. Optim. 34 (4) (2002) 341–354, http://dx.doi.org/10.1080/03052150212723.
[83] T. Ray, K.M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396, http://dx.doi.org/10.1109/TEVC.2003.814902.
[84] J. Zhang, C. Liang, Y. Huang, J. Wu, S. Yang, An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization, Appl. Math. Comput. 211 (2) (2009) 392–416, http://dx.doi.org/10.1016/j.amc.2009.01.048.
[85] A.-R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, J. Global Optim. 35 (4) (2006) 521–549, http://dx.doi.org/10.1007/s10898-005-3693-z.
[86] S. He, E. Prempain, Q.H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optim. 36 (5) (2004) 585–605, http://dx.doi.org/10.1080/03052150410001704854.
[87] S.-F. Hwang, R.-S. He, A hybrid real-parameter genetic algorithm for function optimization, Adv. Eng. Inform. 20 (1) (2006) 7–21, http://dx.doi.org/10.1016/j.aei.2005.09.001.
[88] F.-z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356, http://dx.doi.org/10.1016/j.amc.2006.07.105.
[89] A.D. Belegundu, J.S. Arora, A study of mathematical programming methods for structural optimization, Part I: Theory, Int J. Numer. Methods Eng. 21 (9) (1985) 1583–1599, http://dx.doi.org/10.1002/nme.1620210904.
[90] T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, Eng. Optim. 33 (6) (2001) 735–748, http://dx.doi.org/10.1080/03052150108940941.
[91] C.A. Coello Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Inform. 16 (3) (2002) 193–203, http://dx.doi.org/10.1016/S1474-0346(02)00011-3.