Advances in Engineering Software 195 (2024) 103696

Contents lists available at ScienceDirect

Advances in Engineering Software

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

# Blood-sucking leech optimizer

Jianfu Bai [a], H. Nguyen-Xuan [b], Elena Atroshchenko [c], Gregor Kosec [d], Lihua Wang [e,*], Magd Abdel Wahab [a,d,e,*]

[a] Soete Laboratory, Department of Electrical Energy, Metals, Mechanical Constructions, and Systems, Faculty of Engineering and Architecture, Ghent University, Belgium
[b] CIRTech Institute, HUTECH University, Ho Chi Minh City 700000, Viet Nam
[c] School of Civil and Environmental Engineering, University of New South Wales, Sydney, Australia
[d] Department of Communication Systems, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenija
[e] School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China

ARTICLE INFO

*Keywords:*
Meta-heuristic
Blood-sucking leech optimizer (BSLO)
Benchmark function
Engineering design problem
ANN
Melt electrospinning writing

ABSTRACT

In this paper, a new meta-heuristic optimization algorithm motivated by the foraging behaviour of blood-sucking leeches in rice fields is presented, named Blood-Sucking Leech Optimizer (BSLO). BSLO is modelled by five hunting strategies, which are the exploration of directional leeches, exploitation of directional leeches, switching mechanism of directional leeches, search strategy of directionless leeches, and re-tracking strategy. BSLO and ten comparative meta-heuristic optimization algorithms are used for optimizing twenty-three classical benchmark functions, CEC 2017, and CEC 2019. The strong robustness and optimization efficiency of BSLO are confirmed via four qualitative analyses, two statistical tests and convergence curves. Furthermore, the superiority of BSLO for real-world problems under constraints is demonstrated using five classical engineering problems. Finally, a BSLO-based Artificial Neural Network (ANN) predictive model for diameter prediction of melt electrospinning writing fibre is proposed, which further verifies BSLO's applicability for real-world problems. Therefore, BSLO is a potential optimizer for optimizing various problems. Source codes of BSLO are publicly available at https://www.mathworks.com/matlabcentral/fileexchange/163106-blood-sucking-leech-optimizer.

## 1. Introduction

Optimization aims to search for the maximum or minimum value, and algorithms used for this search process are called optimization algorithms, which are roughly deterministic methods and meta-heuristic algorithms [1]. Deterministic methods can search for an exact optimal solution for small-scale and simple problems. These methods are easy to trap into local optimal solutions, require the derivative information of objective functions, and are time-consuming. As science and society develop constantly, more large-scale, and multi-dimensional problems need to be optimized [2]. However, it is a challenge for deterministic methods to find optimal solutions for these problems despite consuming huge computational costs. In contrast, a meta-heuristic optimization algorithm is regarded as a more efficient method for solving large-scale, and multi-dimensional problems, and has been extensively researched [3,4]. Meta-heuristic algorithms show a better ability of local optima avoidance for complex problems, which are owing to the simple position update with the random operator. Moreover, a meta-heuristic algorithm can be regarded as a black box, so doesn't require the derivative information, and only focuses on the input and output. Hence, meta-heuristic algorithms are popularly used for optimization problems from diverse fields.

Meta-heuristic optimization algorithms search for the optimal solution via a method called "trial and error", and are classified into four categories via various inspirations: Evolutionary Algorithm (EA), Physics-Based Algorithm (PhA), Human-Based Algorithm (HBA), and Swarm Intelligence (SI) Algorithm [5]. EAs are from the inspiration of natural law of evolution, among which Genetic Algorithm (GA) motivated by the Darwinian theory of species evolution is the most popular EA [6-8]. The candidate solution in GA is regarded as chromosome, in which every variable is treated as a gene. GA updates the chromosome to search for the optimal solution using selection, crossover, and mutation strategies. Unlike GA, which selects the same number of next generation population from their parents with good fitness values, Evolution Strategy (ES) generates a temporary population with a different number of individuals from their parents using fitness values or other strategies

* Correspondence
*E-mail addresses:* lhwang@tongji.edu.cn (L. Wang), magd.abdelwahab@ugent.be (M. Abdel Wahab).

https://doi.org/10.1016/j.advengsoft.2024.103696
Received 9 March 2024; Received in revised form 29 May 2024; Accepted 2 June 2024
Available online 14 June 2024

**Table 1**
Partial SI algorithms.

| Name | Inspiration | Year |
|---|---|---|
| Ant Lion Optimizer (ALO) [53] | Antlions | 2015 |
| Moth-Flame Optimization (MFO) [54] | Moths | 2015 |
| Salp Swarm Algorithm (SSA) [55] | Salps | 2017 |
| Grasshopper Optimisation Algorithm (GOA) [56] | Grasshopper swarms | 2017 |
| Sunflower Optimization Algorithm (SFO) [57] | Sunflowers | 2018 |
| Manta Ray Foraging Optimization (MRFO) [58] | Manta rays | 2020 |
| Balancing Composite Motion Optimization [59] | Solution space | 2020 |
| New Caledonian crow learning algorithm (NCCLA) [60] | New Caledonian crows | 2020 |
| Orca Predation Algorithm (OPA) [61] | Orcas | 2021 |
| Artificial Hummingbird Algorithm (AHA) [62] | Hummingbirds | 2021 |
| Honey Badger Algorithm (HBA) [63] | Honey badgers | 2021 |
| Red Fox Optimization Algorithm (RFO) [64] | Fox | 2021 |
| Artificial Rabbits Optimization (ARO) [65] | Rabbits | 2022 |
| Gannet Optimization Algorithm (GOA) [66] | Gannets | 2022 |
| Coati Optimization Algorithm (COA) [67] | Coatis | 2022 |
| Starling Murmuration Optimizer (SMO) [68] | Starlings | 2022 |
| Termite Life Cycle Optimizer (TLCO) [69] | The termite colony | 2022 |
| Fire Hawk Optimizer (FHO) [70] | The hawks | 2023 |
| Meerkat Optimization Algorithm (MOA) [71] | Meerkats | 2023 |
| Genghis Khan Shark Optimizer (GKSO) [72] | The Genghis Khan shark | 2023 |
| Walrus optimizer (WO) [73] | Walruses | 2023 |

[9]. The size of the temporary population is controlled by some parameters, and then crossover and mutation are performed in this population. Other EAs are Evolutionary Programming (EP) [10], Differential Evolution (DE) [11], Backtracking Search Optimization Algorithm (BSA) [12], and Forest Optimization Algorithm (FOA) [13].

The second kind of meta-heuristics is PhA inspired by physical law in universe. A classical representation of PhA is Simulated Annealing (SA) based on annealing principle [14]. In SA, the current optimal solution has a low probability of being replaced by a candidate solution with a bad fitness value, which improves the SA's ability of local optima avoidance. The low probability is controlled by temperature, and gradually decreases to zero with decreasing temperature. Other examples are the Gravitational Search Algorithm (GSA) [15], Flow Regime Algorithm (FRA) [16], Archimedes Optimization Algorithm (AOA) [17], Lichtenberg Algorithm (LA) [18], Flow Direction Algorithm (FDA) [19], Fick's Law Optimization (FLA) [20], Snow Ablation Optimizer (SAO) [21], and Kepler Optimization Algorithm (KOA) [22].

The third category is HBA, which is from the inspiration by the production and life of human society. Teaching-Learning-Based Optimization (TLBO) is the typical representative among HBA [23]. TLBO imitates the effect of teachers on students, in which the optimal solution is obtained using two phases of "Teacher" and "Student". There are many other HBAs, such as Social Engineering Optimizer (SEO) [24], Poor and Rich Optimization (PRO) [25], Heap-Based Optimizer (HBO) [26], Group Teaching Optimization Algorithm (GTOA) [27], Past Present Future (PPF) [28], Driving Training-Based Optimization (DTBO) [29], Boxing Match Algorithm (BMA) [30] and Special Forces Algorithm (SFA) [31].

The final SI algorithms, motivated by the hunting behaviour, hierarchy, and reproductive behaviour of plants and animals, are the most widely studied and applied algorithms among meta-heuristic algorithms [32-36]. The most well-known SI algorithms are Particle Swarm Algorithm (PSO) motivated by the collection phenomenon of school of fish and bird [37], Ant Colony Optimization (ACO) based on the ants' behaviour of releasing pheromones [38], and Artificial Bee Colony (ABC) motivated by the cooperative phenomenon of employed bees, onlookers and scouts [39] ABC has been popularly researched and utilized to different fields, like the parameter optimization of artificial neural networks and support vector machine, image segmentation, data classification, data mining and scheduling problem [40-42]. Another very popular SI algorithm is Grey Wolf Optimizer (GWO) motivated by the hierarchical phenomenon in the hunting process of grey wolves, which has been widely applied to machine learning, engineering design, wireless sensor network, medical and bioinformatics, and image processing application [43]. Grey wolves are divided into $\alpha$, $\beta$, $\delta$ and $\omega$ wolves, among which $\alpha$, $\beta$, $\delta$ wolves are the leadership hierarchy. The central position of $\alpha$, $\beta$, $\delta$ wolves is regarded as the position of prey, and $\omega$ wolves follow the three leading wolves to search for prey. Currently, many SI algorithms are presented, like the Squirrel Search Algorithm (SSA) motivated by hunting behavior of southern flying squirrels [44], Chimp Optimization Algorithm (ChOA) inspired by the individuals' diversity and sexual behaviour of chimps [45], Slime Mould Algorithm (SMA) imitating the diffusion and foraging behaviours of slime mould [46], Black Widow Optimization Algorithm (BWO) motivated by the mating phenomenon of black widow spiders [47], Golden Eagle Optimizer (GEO) inspired the spiral hunting trajectory of golden eagles [48], African Vultures Optimization Algorithm (AVOA) based on the navigation and hunting phenomenon of African vultures [49], White Shark Optimizer (WSO) motivated by the exceptional senses of white sharks [50], Snake Optimizer (SO) imitating a snakes' especial mating phenomenon [51], and Greylag Goose Optimization (GGO) motivated by the three behaviours of crayfish including summer resort, competition and foraging [52]. Other such algorithms are presented in Table 1.

The searching process of SI algorithms is classified into exploration and exploitation [74]. SI algorithms search for the optimal solution using exploration and exploitation after initializing population and obtaining the first optimal solution. Exploration means that candidate solutions search for the whole solution space with large steps, while exploitation means that the potential space obtained by the exploration phase is exploited by candidate solutions with small steps. The exploration and exploitation of SI algorithms are relatively contradictory. Excessive exploration can lead to insufficient exploitation, which will lead to low accuracy of the search solution, while over-exploitation can cause the candidate solution to trap into local optima. Therefore, how to find a balance strategy between exploration and exploitation remains a huge challenge [75]. Chen et al. published a modified Whale Optimization Algorithm (WOA) algorithm via a chaos mechanism based on quasi-opposition, which improved the convergence and balance of WOA [76]. Another improved WOA was also introduced by adding a new parameter $\beta$, random movement, an inertia weight and modified the co-efficient vectors $A$ and $C$, which also enhanced the balance of exploration and exploitation [77]. Similarly, GWO's imbalance between exploration and exploitation was also investigated by many researchers [78]. For example, Yu et al. introduced an Opposition-based learning Grey Wolf Optimizer (OGWO), which enhanced the performance of local optimal avoidance using opposition-based learning approach and improved GWO's balance performance between exploration and exploitation via modifying parameter $\overrightarrow{a}$ [79]. Also, Jiang et al. provided another balance mechanism of exploration and exploitation by dividing GWO's hunting process into two phases, and combined it with group-stage competition mechanism to enhance GWO's performance [80]. Dong et al. introduced an improved ALO, which provided ALO a nice balance strategy by adding dynamic opposite learning [81]. Si et al. used generation jumping and opposition-based initialization strategies to balance SSA's exploration and exploitation, obtained good results [82]. Mostafa et al. enhanced Gorilla Groups Optimizer (GTO) using three methods: the elite Opposition Based-Learning, tangent Flight and Cauchy Inverse Cumulative Distribution Operator, which provided GTO a better balancing strategy [83]. Owing to SO's insufficient diversity and imbalance between exploration and exploitation, Yao et al. successfully enhanced SO's performance using a novel opposition-based learning method, Tent-chaos & Cauchy mutation, sine-cosine composite disturbance, and dynamic parameters [84]. For the purpose of solving the insufficient performance of Golden Jackal Optimizer for robot path planning, pre-deceasing slow nonlinear energy decay method was introduced into Golden Jackal Optimizer to improve its balance ability between exploration and exploitation [85]. Also, its capacity of local

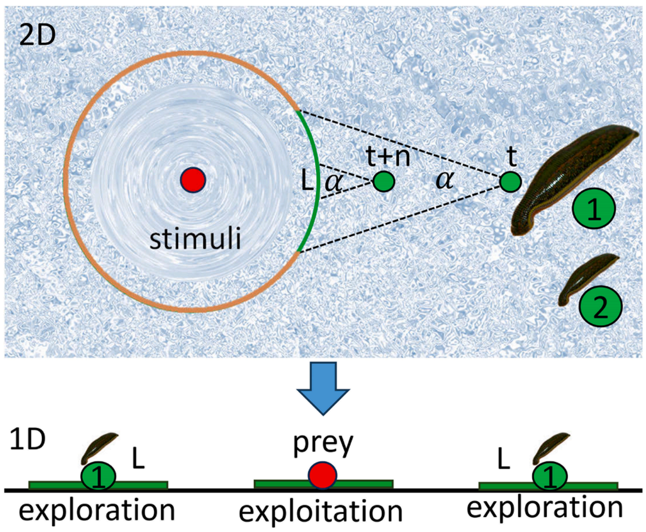


**Fig. 1.** The hunting process of blood-sucking leeches.

optima avoidance was enhanced by adding roulette wheel selection strategy. Although new balance strategies are constantly proposed, balancing exploration and exploitation is still a thorny problem. Therefore, an optimization algorithm based on Balancing Composite Motion (BCMO) [59] presents itself as a suitable candidate due to its ability to balance global and local search mechanisms.

Furthermore, no one algorithm can optimize all problems according to No Free Lunch Theorem for Optimization [86]. In other words, one algorithm can achieve great efficiency for some problems, but it may obtain unsatisfactory results for other problems, which encourages the emergence of novel algorithms or modify existing algorithms for various problems. Considering the two challenges mentioned, in this study, we propose a new SI algorithm inspired by the foraging behaviour of the blood-sucking leech in rice fields to give a novel balance method between exploration and exploitation, named Blood-Sucking Leech Optimizer (BSLO). When blood-sucking leeches are stimulated, most leeches will swim toward humans at a small angle and few leeches will lose the direction of humans. In addition, leeches will be randomly thrown into rice fields and search for humans again when humans feel the bites of leeches. According to these phenomena, a mathematical model of BSLO is developed for the first time. The superior performance of BSLO is demonstrated using twenty-three classical benchmark functions, CEC 2017, CEC 2019 and five classical engineering design problems. Finally, BSLO is used for the Artificial Neural Network (ANN) predictive model to predict the fibre diameter of melt electrospinning writing, a micro-nano fibre manufacturing technology [87,88].

The sections are described as follows. Section 2 explains the inspiration and mathematical formulation of BSLO. Section 3 presents the optimization results of BSLO and ten comparative algorithms using three types of evaluation functions mentioned. The optimization results of BSLO and ten comparative algorithms using five classical engineering design problems are shown in Section 4. The BSLO-based ANN predictive model for melt electrospinning writing is presented in Section 5. The final section presents the conclusion and the future work.

## 2. BSLO

### 2.1. Inspiration

Leeches, a widely researched animal for medical treatment over thousands of years, are still applied for the modern medical treatment, such as cardiovascular diseases, reconstructive and microsurgery, cancer and metastasis, diabetes mellitus and its complications, and infectious diseases [89]. Thousands of species of leeches live in the world, among which the blood-sucking leech prefers to live in freshwater, such as rice fields. The blood-sucking leech is very common in Chinese rice fields and feed on the blood of vertebrates, including humans. The blood-sucking leech has many receptors, such as mechanoreceptors and chemoreceptors [90–92]. When the blood-sucking leech contacts the water wave, the gradient in temperature, and the waterborne odors of food, it can continually track prey using its receptors. If the prey moves, the blood-sucking leech will orient to the new position of the prey. In addition, most leeches can swim toward prey at a slight angle when they feel these stimuli, and only a small number of leeches will lose the direction [93]. Hence, the blood-sucking leech can efficiently orient prey and catch prey. When the blood-sucking leech bites human feet in rice fields and sucks their blood for a while, then it will be removed by humans and randomly thrown back into the rice fields. These leeches will search for humans again. BSLO is inspired by these foraging behaviours of the blood-sucking leech, and the mathematical model of BSLO will be presented in the following sections.

### 2.2. Initialization

As the same as other SI algorithms, BSLO first initializes a set of stochastically distributed candidate solutions in the solution space. Every candidate solution represents a blood-sucking leech, as shown in Eq. (1):

$$X = [x_1,\ x_2, \cdots,\ x_D] \tag{1}$$

where $D$ denotes the problem dimension. For minimization problems, all blood-sucking leeches are shown in Eq. (2), among which the blood-sucking leech obtaining the smallest fitness value is regarded as the optimal solution in current iteration. The history optimal solution is replaced by the current optimal solution with a smaller fitness value.

$$X_{all} = \begin{array}{c} \begin{array}{ccccc} x_{1,1} & \cdots & x_{1,j} & x_{1,D-1} & x_{1,D} \end{array} \\ \begin{bmatrix} x_{1,2} & \cdots & x_{2,j} & \cdots & x_{2,D} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,D} \end{bmatrix} \\ \begin{array}{ccccc} x_{N,1} & \cdots & x_{N,j} & x_{N,D-1} & x_{N,D} \end{array} \end{array} \tag{2}$$

where $X_{all}$ represents blood-sucking leeches, initialized by Eq. (3). $x_{i,j}$ denotes the $j_{th}$ position of $i_{th}$ of the blood-sucking leech, and $N$ indicates the number of the blood-sucking leeches:

$$X_{all} = rand(N, D) \times (ub - lb) + lb \tag{3}$$

where *rand* takes a random value from [0, 1], and *ub* and *lb* indicate the upper and lower bound of optimization problems, respectively.

### 2.3. Mathematical model of BSLO

The foraging behaviour of blood-sucking leeches is described in Fig. 1. Leeches are divided into two types including the directional and directionless leeches, as shown in Eq. (4) and Eq. (5), respectively. The directional leeches can swim toward prey at $\alpha^{\circ}C$ when these leeches contact with circular wave stimuli produced by humans. Other directionless leeches randomly swim in the search space. After biting humans for a while, leeches will be randomly thrown back into the rice fields by humans and search for humans again.

$$N_1 = floor\left(N \times \left(m + (1 - m) \times \left(\frac{t}{T}\right)^2\right)\right) \tag{4}$$

$$N_2 = N - N_1 \tag{5}$$

where $N_1$ and $N_2$ are the number of the directional leeches and the

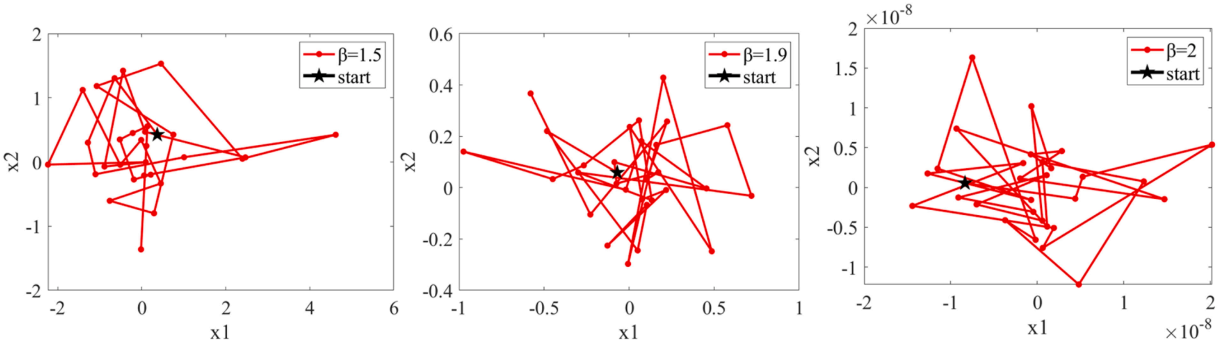


**Fig. 2.** The Lévy movement of BSLO for different $\beta$s.

directionless leeches, respectively. *floor* is a function in MATLAB for rounding down. $T$ and $t$ indicate the maximum and current iterations, respectively. $m$ represents the ratio parameter, which decides on the number of the directional leeches and the directionless leeches and is set to 0.8 in this paper. Hence, most leeches can search directionally for humans at the beginning due to their foraging behaviour. Also, a number of them increases with increasing iterations since more and more leeches can find humans.

#### 2.3.1. Exploration strategy of directional leeches

When blood-sucking leeches feel stimuli, such as the water wave, $N_1$ leeches can swim toward humans at a slight angle. As shown in Fig. 1, "$t$" and "$t+n$" denote different positions of a leech at different iterations, and the green arc length represents the region where the same leech from "$t$" and "$t+n$" position swims to the water ripples. Therefore, the arc length $L$ decreases when leeches gradually approach humans. As iterations increase, leeches gradually approach the global optimum, and the arc length $L$ approaches 0. The arc length $L$ is projected into one dimension, the green $L$ represents the next possible position of leeches. Leeches can search in the region $L$ far away from humans, which is considered as the exploration phase of leeches. The exploration of leeches is shown as Eq. (6).

$$X_{(i,j)}^{t+1} = \begin{cases} X_{(i,j)}^{t} + W_1 \times X_{(i,j)}^{t} - L_1 \ rand < a \ \& \ \left|Prey_{(j)}\right| > \left|X_{(i,j)}^{t}\right| \\ X_{(i,j)}^{t} + W_1 \times X_{(i,j)}^{t} + L_1 \ rand < a \ \& \ \left|Prey_{(j)}\right| < \left|X_{(i,j)}^{t}\right| \\ X_{(i,j)}^{t} + W_1 \times X_{(i,k)}^{t} - L_2 \ rand > a \ \& \ \left|Prey_{(j)}\right| > \left|X_{(i,j)}^{t}\right| \\ X_{(i,j)}^{t} + W_1 \times X_{(i,k)}^{t} + L_2 \ rand > a \ \& \ \left|Prey_{(j)}\right| < \left|X_{(i,j)}^{t}\right| \end{cases} \tag{6}$$

where $t$ denotes the current iteration, $X_{(i,j)}^{t}$ and $X_{(i,j)}^{t+1}$ are the $j_{th}$ position of the $i_{th}$ leech in the current and next iterations, respectively. $k$ is the random integer in the $[1, D]$. $X_{(i,k)}^{t}$ represents the random $k_{th}$ position of the $i_{th}$ leech. The latter two formulas of $X_{(i,j)}^{t+1}$ provide a connection between the current dimension and other dimensions, which means that other dimensional position of the $i_{th}$ leech can affect the next current dimensional position of the $i_{th}$ leech. The method can increase the diversity during exploration phase, but the main impact on the next $j_{th}$ position of the $i_{th}$ leech is still the current dimensional position. Therefore, $a$ is very close to 1, set to 0.97 in this paper. Here, *rand* is the random value in the range [0, 1]. When *rand* is smaller than $a$, $X_{(i,j)}^{t+1}$ is calculated by the first two formulas. Otherwise, $X_{(i,j)}^{t+1}$ is calculated by the latter two formulas. $Prey_{(j)}$ represents the $j_{th}$ position of the optimal solution obtained so far. $W_1$ is a tiny disturbance coefficient for leeches, which also increases the diversity during exploration phase, given as Eq. (7). $L_1$ and $L_2$ are two sizes of $L$, calculated by Eq. (8) and Eq. (9). When the next $j_{th}$ position of the $i_{th}$ leech is affected by current dimensional position, $L_1$ is selected. Otherwise, $L_2$ is selected.

$$W_1 = b \times \left(1 - \frac{t}{T}\right) \times LV_{(i,j)} \tag{7}$$

where $b$ is a small value to ensure that the disturbance coefficient $W_1$ is a very small, set to 0.001. $LV_{(i,j)}$ is a random vector calculated by Eq. (10), which can provide the random disturbance.

$$L_1 = r_1 \times \left|Prey_{(j)} - X_{(i,j)}^{t}\right| \times PD \times \left(1 - \frac{k_2}{N}\right) \tag{8}$$

$$L_2 = \left|Prey_{(j)} - X_{(i,k)}^{t}\right| \times PD \times \left(1 - r_1^2 \times \frac{k_2}{N}\right) \tag{9}$$

where $r_1$ denotes a random value in the $[-1, 1]$, $k_2$ is the random integer, given from $\left[1, floor\left(N \times \left(1 + \frac{t}{T}\right)\right)\right]$. $PD$, called the perceived distance, is used to imitate the distance perceived by leeches from humans, and is calculated by Eq. (12).

$$LV_{(i,j)} = 0.5 \ \times l\acute{e}vy(y) \tag{10}$$

$$L\acute{e}vy(y) = 0.01 \ \times \frac{\mu \times \sigma}{\left|v^{\left(\frac{1}{\beta}\right)}\right|}; \ \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times \left(2^{\frac{\beta-1}{2}}\right)}\right)^{1/\beta} \tag{11}$$

where $L\acute{e}vy(y)$ is the levy flight distribution function, $\mu$ and $v$ take random values in the [0, 1], $\sigma$ is calculated based on the second expression in Eq. (11), while $\beta$ is calculated by Eq. (14).

$$PD = s \times \left(1 - \frac{t}{T}\right) \times r_2 \tag{12}$$

$$s = \begin{cases} 8 - \left(-\left(\frac{t}{T}\right)^2 + 1\right) \ if \ rand < 0.5 \\ 8 - 7 \ \times \left(-\left(\frac{t}{T}\right)^2 + 1\right) \ otherwise \end{cases} \tag{13}$$

where $r_2$ is random values in the $[-1, 1]$.

$$\beta = -0.5 \ \times \left(\frac{t}{T}\right)^6 + \left(\frac{t}{T}\right)^4 + 1.5 \tag{14}$$

where $\beta$ increases from 1.5 to 2 with increasing iterations, which can decrease the values of $L\acute{e}vy(y)$ as shown in Fig. 2. Hence, leeches can search for the space solution with big steps at the beginning and exploit the potential region with small steps in the later iterations, which helps leeches search for humans.

#### 2.3.2. Exploitation strategy of directional leeches

SI algorithms are divided into exploration and exploitation phases, and the proposed BSLO also follows this rule. After exploring the search

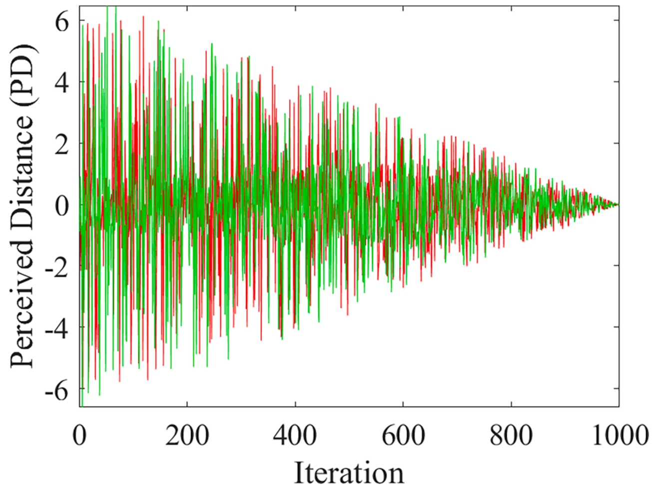


**Fig. 3.** The values of perceived distance with 1000 iterations over 2 runs.

space for many iterations, $N_1$ leeches gradually approach humans and are exposed to more intense stimuli, and then obtain the potential region of humans. After that, leeches enter the exploitation phase in the $L$ region close to humans, which is calculated by Eq. (15).

$$X_{(i,j)}^{t+1} = \begin{cases} Prey_{(j)} + W_1 \times Prey_{(j)} - L_3 & rand < a \ \& \ \left|Prey_{(j)}\right| > \left|X_{(i,j)}^t\right| \\ Prey_{(j)} + W_1 \times Prey_{(j)} + L_3 & rand < a \ \& \ \left|Prey_{(j)}\right| < \left|X_{(i,j)}^t\right| \\ Prey_{(j)} + W_1 \times Prey_{(j)} - L_4 & rand > a \ \& \ \left|Prey_{(j)}\right| > \left|X_{(i,j)}^t\right| \\ Prey_{(j)} + W_1 \times Prey_{(j)} + L_4 & rand > a \ \& \ \left|Prey_{(j)}\right| < \left|X_{(i,j)}^t\right| \end{cases} \tag{15}$$

where $W_1$ is a very small disturbance coefficient for prey, which is also calculated by Eq. (7). The difference between the value of $W_1$ and the exploration phase is that $b$is based on iterations. When $t < 0.1 \times T$, the value of $b$ is the same as in the exploration phase. After many iterations ($t \geq 0.1 \times T$), leeches find the potential region. The smaller disturbance coefficient is required to search for the optimal solution more accurately. Hence, $b$ is set to 0.00001. $L_3$ and $L_4$ are two sizes of $L$, calculated by Eq. (16) and Eq. (17). As mentioned in the exploration phase, the current dimensional position is still the main factor to affect the next $j_{th}$ position of the $i_{th}$ leech in the exploitation phase. Hence, $L_3$ is selected when $rand < a$, otherwise, $L_4$ is selected:

$$L_3 = \left|Prey_{(j)} - X_{(i,j)}^t\right| \times PD \times \left(1 - r_3 \times \frac{k_2}{N}\right) \tag{16}$$

$$L_4 = \left|Prey_{(j)} - X_{(i,k)}^t\right| \times PD \times \left(1 - r_3 \times \frac{k_2}{N}\right) \tag{17}$$

where $r_3$ is the random value in the range $[-1, 1]$ and $PD$ is calculated by Eq. (12).

### 2.3.3. Switching mechanism of directional leeches

Exploration and exploitation require a switch mechanism. The

**Algorithm 1**
Pseudo-code of BSLO

Initialize BSLO parameters $t$, $t_1$, $t_2$, $m$, $a$, $b$, etc.
Initialize the population $X_{all}$ using Eq. (3).
**While** ($t \leq T$) **do**
Evaluate the Fitness values of $X_{all}$.
Find the optimal solution so far.
**if** $t > t_1$ **then**
/***Re-tracking strategy***/
if $F(X_i^t) = F(Prey^{t-t_2})$ then
Update re-tracking solutions using Eq. (20).
**end if**
**end if**
Update $N_1$ and $N_2$ using Eq. (4) and (5).
Update $s$ and $LV_{(i,j)}$using Eq. (13) and (10).
Update $k$, $k_2$ of $X_{all}$.
/***Search strategies of directional leeches***/
**for** ($i = 1$ to $N_1$) **do**
**for** ($i = 1$ to $D$) **do**
Calculate $PD$ using Eq. (12).
**if** $PD \geq 1$ **then**
/***Exploration phase***/
Update the positions of directional leeches using Eq. (6).
**else**
/***Exploitation phase***/
Update the positions of directional leeches using Eq. (15).
**end if**
**end for**
**end for**
/***Search strategies of directionless leeches***/
**for** ($i = N_2$ to $N$) **do**
**for** ($i = 1$ to $D$) **do**
Update the positions of directionless leeches using Eq. (18).
**end for**
**end for**
$t = t + 1$.
**end while**
Return the optimal solution.

**Table 2**
Twenty-three classical benchmark functions

| Function | Dimension | 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 |
| $F_3(x) = \sum_{i=1}^{d} \left( \sum_{j-1}^{i} x_j \right)^2$ | 30 | [-100, 100] | 0 |
| $F_4(x) = max\{\lvert x_i \rvert, 1 \le i \le n\}$ | 30 | [-100, 100] | 0 |
| $F_5(x) = \sum_{i=1}^{n-1} \left[ 100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2 \right]$ | 30 | [-30, 30] | 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 |
| $F_8(x) = \sum_{i=1}^{n} - x_i \sin\left( \sqrt{\lvert x_i \rvert} \right)$ | 30 | [-500, 500] | -418.9829 × n |
| $F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 30 | [-5.12, 5.12] | 0 |
| $F_{10}(x) = -20exp\left( -0.2\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - exp\left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$ | 30 | [-32, 32] | 0 |
| $F_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left( \frac{x_i}{\sqrt{i}} \right) + 1$ | 30 | [-600, 600] | 0 |
| $F_{12}(x) = \frac{\pi}{n} \left\{ 10sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ <br> $y_i = 1 + \frac{x_i + 1}{4}$ <br> $u(x_i, k, a, m) = \{ \begin{matrix} k(x_i - a)^m \ x_i > a \\ 0 \ -a < x_i < a \\ k(-x_i - a)^m \ x_i < -a \end{matrix}$ | 30 | [-50, 50] | 0 |
| $F_{13}(x) = 0.1\left\{ sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + sin^2(2\pi x_n)] \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | 30 | [-50, 50] | 0 |
| $F_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$ | 2 | [-65.53, 65.53] | 1 |
| $F_{15}(x) = \sum_{i=1}^{11} \left( a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right)^2$ | 4 | [-5, 5] | 0.0003 |
| $F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$ | 2 | [-5, 5] | -1.0316 |
| $F_{17}(x) = \left\{ x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right\}^2 + 10\left( 1 - \frac{1}{8\pi} \right) cosx_1 + 10$ | 2 | [-5, 10] × [0, 15] | 0.398 |
| $F_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right] \times \left[ 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]$ | 2 | [-5, 5] | 3 |
| $F_{19}(x) = -\sum_{i=1}^{4} c_i \ exp\left( -\sum_{j=1}^{3} a_{ij} \left( x_j - p_{ij} \right)^2 \right)$ | 3 | [0, 1] | -3.86 |
| $F_{20}(x) = -\sum_{i=1}^{4} c_i \ \exp\left( -\sum_{j=1}^{6} a_{ij} \left( x_j - p_{ij} \right)^2 \right)$ | 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 |

**Table 3**
Parameters$m$, $a$, $b$, $b_2$, $t_1$, and $t_2$.

| Parameter | Low | Medium | High |
|---|---|---|---|
| $m$ | 0.7 | 0.8 | 0.9 |
| $a$ | 0.96 | 0.97 | 0.98 |
| $b$ | 0.0005 | 0.0010 | 0.0015 |
| $b_2$ | 0.000005 | 0.000010 | 0.000015 |
| $t_1$ | 10 | 20 | 30 |
| $t_2$ | 10 | 20 | 30 |

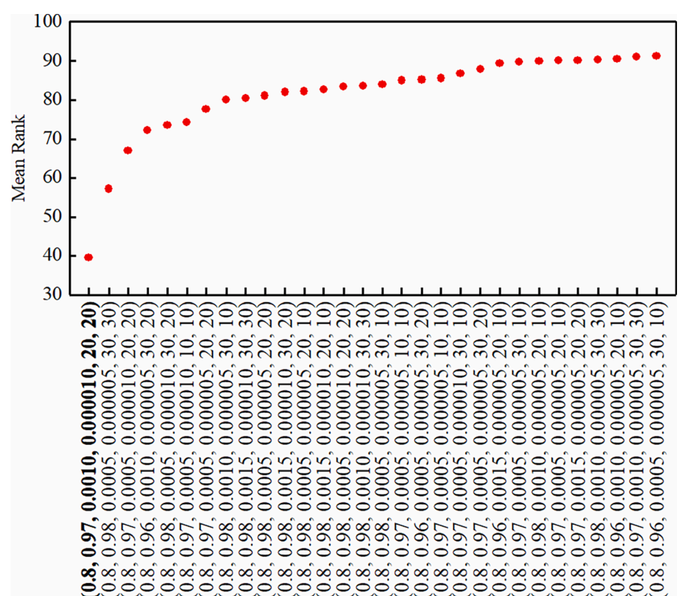


**Fig. 4.** The top 30 optimization results of BSLO using different parameter settings.

perceived distance $PD$ is designed to imitate the distance perceived by leeches from humans. Most directional leeches explore humans at the beginning. Hence, most leeches sense that they are far away from humans at the beginning, so most values of $PD$ are large. Few values of $PD$ are small at the beginning since some leeches are closer to humans after initialization. To totally imitate this situation, $PD$ is calculated by Eq. (12), where Eq. (12) is based on two scenarios as shown in Eq. (13). With increasing iterations, more and more leeches can finally find or become close to the optimal solution. Therefore, $PD$ is gradually close to zero. When $PD > 1$, leeches perceive that they are far away from humans and BSLO enters the exploration phase. Otherwise, leeches perceive that they are closer to humans and BSLO enters the exploitation phase. The values of $PD$ in the iterations are shown in Fig. 3. It can be seen that

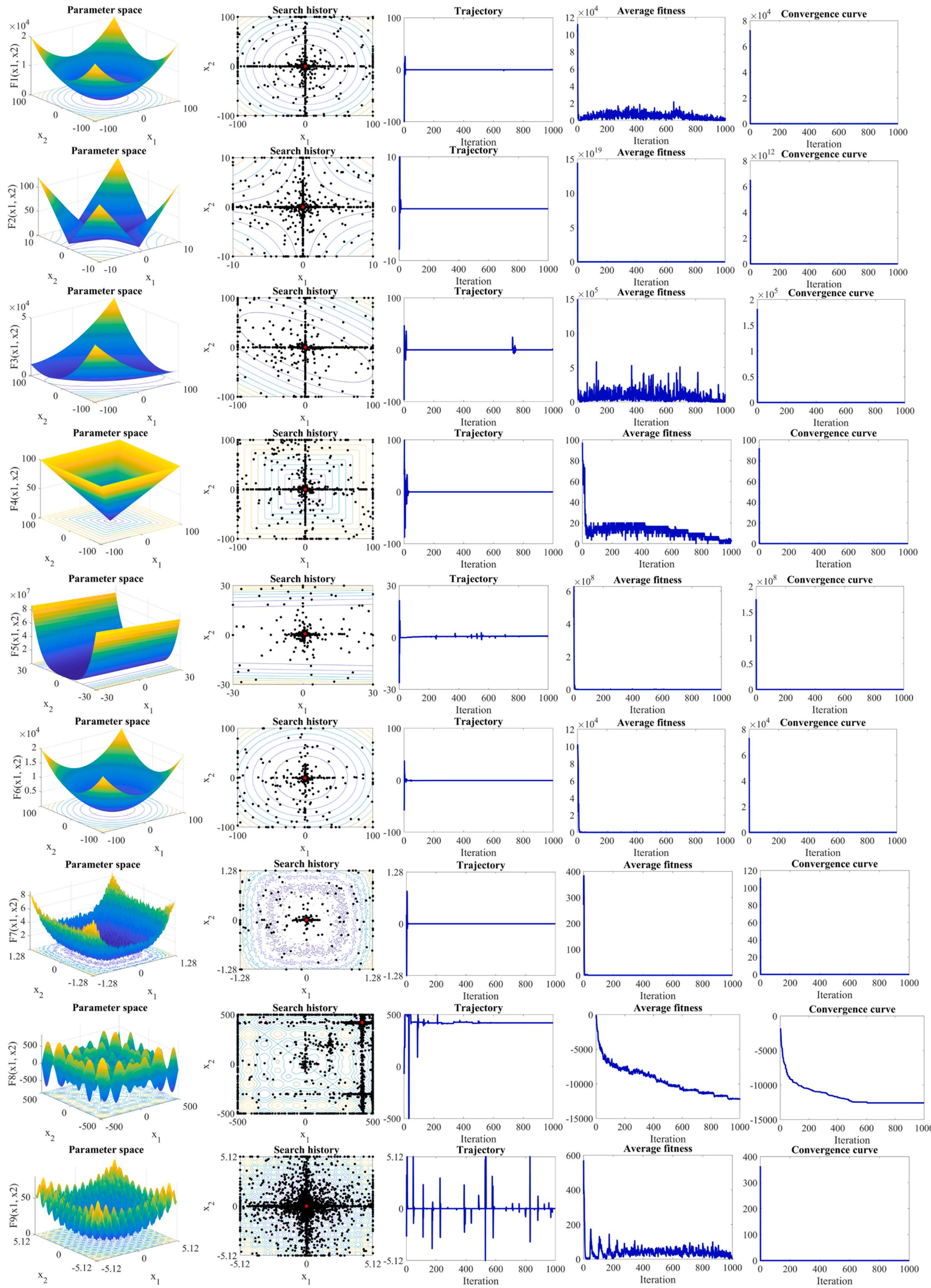


**Fig. 5.** Qualitative analysis for the 23 classical benchmark functions.

directional leeches mainly perform exploration at the beginning and exploitation in the later iterations. Also, exploitation exists in the entire iterations.

### 2.3.4. *Search strategy of directionless leeches*

After feeling stimuli, a few leeches ($N_2$) misjudge information and swim to the wrong direction. As increasing iterations, more and more

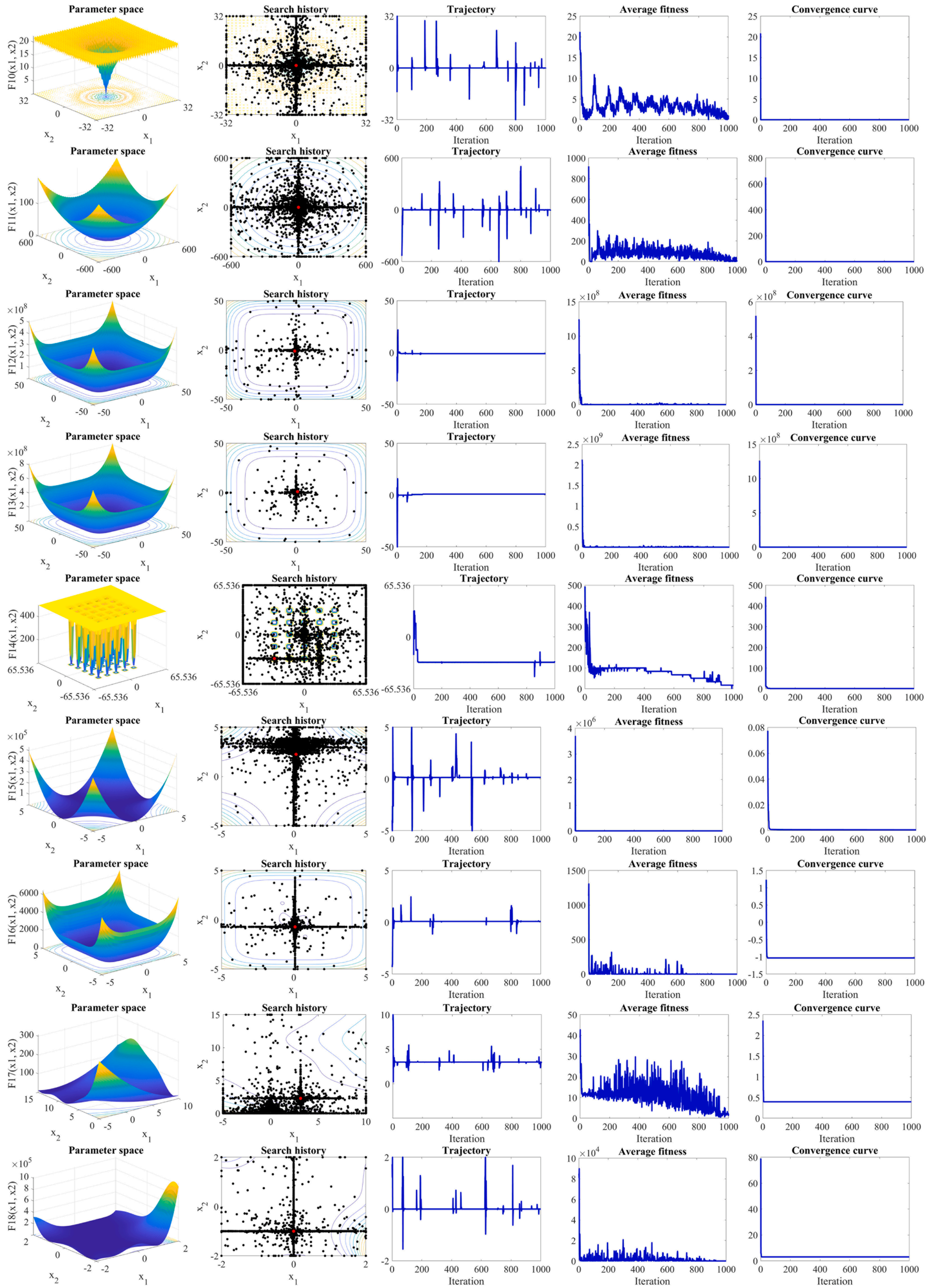


**Fig. 5.** (*continued*).

leeches find or are close to humans, and $N_2$ leeches are gradually close to zero. A distribution function called Lévy flight is added to design the mathematical model of directionless leeches. Hence, directionless leeches randomly search for humans in rice fields, which increases BSLO's diversity and its ability of local optima avoidance. In addition, directionless leeches can wander randomly around themselves and away from humans, or around humans. The next positions of these leeches are obtained by Eq. (18).

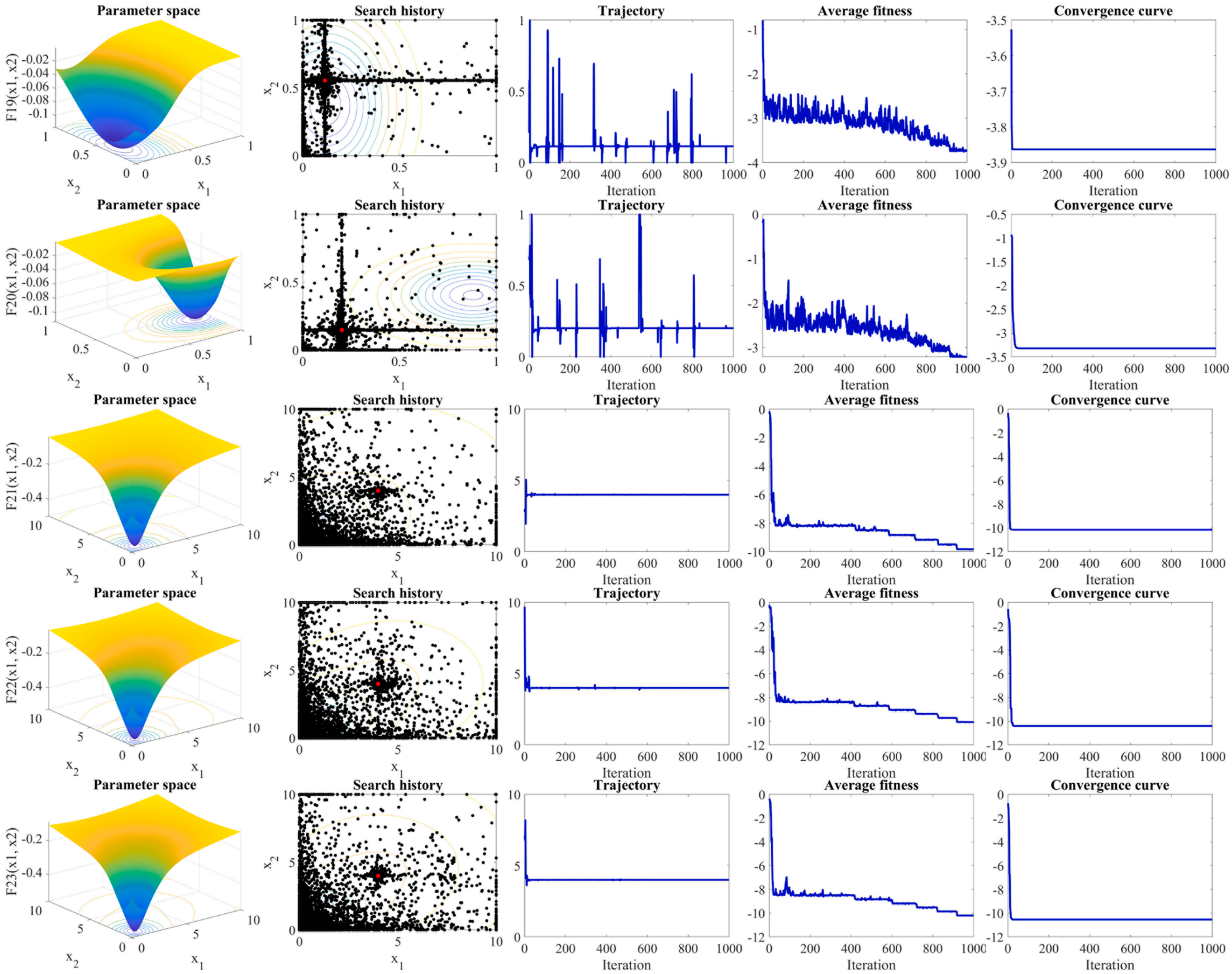

**Fig. 5.** (*continued*).

**Table 4**
Important parameter information for BSLO and comparative algorithms.

| Algorithm | Parameter | Value |
|---|---|---|
| BSLO | $t_1$, $t_2$, $m$, $a$, $b$, $b_2$ | 20, 20, 0.8, 0.97, 0.001, 0.00001 |
| PSO | $w$ | 0.6 |
| | $c_1$ | 0.5 |
| | $c_2$ | 0.6 |
| | Velocity limit | 10% of dimension range |
| GWO | $a$(convergence parameter) | Decreased linearly from 2 to 0 |
| WOA | $a$ | Decreased linearly from 2 to 0 |
| | $b$ | 1 |
| SHO | $h$ | Decreased linearly from 5 to 0 |
| | $\overrightarrow{M}$(random vector) | [0.5, 1] |
| AOA | $\alpha$(sensitive parameter) | 5 |
| | $u$ (control parameter) | 0.5 |
| AO | $\alpha$ | 0.1 |
| | $\delta$ | 0.1 |
| GJO | $c_1$(constant) | 1.5 |
| | $\beta$(constant) | 1.5 |
| SCSO | $r_G$(Sensitivity range) | Decreased linearly from 2 to 0 |
| | $R$(transitions control parameter) | $[-2r_G, 2r_G]$ |
| FOX | $c_1$ | 0.18 |
| | $c_2$ | 0.82 |
| DO | $\alpha$ | [0,1] |
| | $k$ | [0,1] |

$$X_{(i,j)}^{t+1} = \begin{cases} \frac{t}{T} \times \left| Prey_{(j)} - X_{(i,j)}^t \right| \times LV2_{(i,j)} \times X_{(i,j)}^t \ if\ rand < 0.5 \\ \frac{t}{T} \times \left| Prey_{(j)} - X_{(i,j)}^t \right| \times LV2_{(i,j)} \times Prey_{(j)}\ otherwise \end{cases} \tag{18}$$

where the selection probabilities for both wandering situations are equal. $LV2_{(i,j)}$ is also Lévy flight distribution function. To prevent wasting too many meaningless distributions on problems with only positive bound, $LV2_{(i,j)}$ is divided two situations, shown in Eq. (19). When problems only have non-negative bound ($lb \geq 0$), $LV2_{(i,j)}$ takes the absolute value of $LV_{(i,j)}$, otherwise, $LV2_{(i,j)}$ is equal to $LV_{(i,j)}$.

$$LV2_{(i,j)} = \begin{cases} \left| LV_{(i,j)} \right| \ if\ lb \geq 0 \\ LV_{(i,j)}\ otherwise \end{cases} \tag{19}$$

### 2.3.5. *Re-tracking strategy*

After entering exploration and exploitation for many times ($t_1$), some leeches successfully find humans and suck the blood. When humans sowing seeds in rice fields feel the sting, they randomly throw the leeches attached to their feet into the rice fields. This process occurs periodically ($t_2$), which means that the optimal solution ($Prey^{t-t_2}$) and the fitness value of $X_i^t$ are compared to see if they are equal. After that, these thrown leeches can search for humans again. $t_1$ and $t_2$ are set to 20 in this paper. These positions of thrown leeches are given by Eq. (19).

$$X_i = rand(1, D) \times (ub - lb) + lb\ t > t_1\ \&\ F\left(X_i^t\right) = F\left(Prey^{t-t_2}\right) \tag{20}$$

**Table 5**
Results of BSLO and other ten algorithms for twenty-three classical test functions (the best average optimal solution is marked in bold).

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Best | 0.000E+00 | 1.053E+03 | 3.759E-62 | 7.074E-167 | 0.000E+00 | 8.721E-290 | 2.451E-305 | 9.984E-120 | 8.972E-250 | 0.000E+00 | 3.046E-09 |
| | Average | **0.000E+00** | **2.203E+03** | 4.736E-59 | 1.782E-153 | **0.000E+00** | 7.633E-80 | 8.940E-219 | 4.532E-113 | 1.004E-228 | **0.000E+00** | 1.587E-08 |
| | STD | 0.000E+00 | 6.062E+02 | 8.670E-59 | 4.777E-153 | 0.000E+00 | 4.167E-79 | 0.000E+00 | 1.398E-112 | 0.000E+00 | 0.000E+00 | 1.113E-08 |
| F2 | Best | 0.000E+00 | 1.104E+01 | 7.434E-36 | 4.082E-118 | 0.000E+00 | 0.000E+00 | 1.305E-151 | 2.572E-68 | 1.096E-130 | 0.000E+00 | 2.363E-05 |
| | Average | **0.000E+00** | **2.329E+01** | 6.669E-35 | 1.980E-104 | **0.000E+00** | **0.000E+00** | 1.853E-106 | 1.819E-66 | 6.623E-121 | **0.000E+00** | 8.490E-05 |
| | STD | 0.000E+00 | 6.519E+00 | 6.871E-35 | 9.051E-104 | 0.000E+00 | 0.000E+00 | 1.015E-105 | 2.792E-66 | 3.578E-120 | 0.000E+00 | 6.327E-05 |
| F3 | Best | 0.000E+00 | 3.009E+03 | 1.482E-20 | 7.633E+02 | 0.000E+00 | 6.454E-250 | 2.277E-298 | 1.789E-51 | 5.667E-223 | 0.000E+00 | 1.461E-01 |
| | Average | **0.000E+00** | **8.146E+03** | 1.064E-15 | 1.902E+04 | **0.000E+00** | 5.994E-04 | 1.558E-201 | 3.041E-38 | 2.753E-204 | **0.000E+00** | 5.707E-01 |
| | STD | 0.000E+00 | 3.081E+03 | 2.979E-15 | 1.072E+04 | 0.000E+00 | 1.973E-03 | 0.000E+00 | 1.613E-37 | 0.000E+00 | 0.000E+00 | 4.194E-01 |
| F4 | Best | 0.000E+00 | 1.408E+01 | 2.705E-16 | 1.359E-02 | 0.000E+00 | 1.803E-90 | 6.475E-153 | 4.551E-36 | 6.477E-110 | 0.000E+00 | 1.818E-02 |
| | Average | **0.000E+00** | **2.277E+01** | 1.369E-14 | 3.111E+01 | **0.000E+00** | 1.579E-02 | 1.321E-117 | 2.823E-33 | 6.399E-100 | **0.000E+00** | 7.577E-02 |
| | STD | 0.000E+00 | 4.038E+00 | 2.117E-14 | 3.218E+01 | 0.000E+00 | 2.078E-02 | 7.038E-117 | 8.144E-33 | 3.471E-99 | 0.000E+00 | 4.736E-02 |
| F5 | Best | 2.489E+01 | 6.478E+04 | 2.504E+01 | 2.617E+01 | 2.869E+01 | 2.647E+01 | 4.606E-07 | 6.080E+00 | 2.605E+01 | 2.869E+01 | 2.472E+01 |
| | Average | 2.522E+01 | 4.357E+05 | 2.663E+01 | 2.695E+01 | 2.880E+01 | 2.810E+01 | **7.451E-04** | 2.754E+01 | 2.805E+01 | 2.876E+01 | 2.598E+01 |
| | STD | 1.337E-01 | 3.755E+05 | 9.130E-01 | 3.426E-01 | 1.173E-01 | 4.831E-01 | 7.925E-04 | 8.071E-01 | 9.879E-01 | 3.486E-02 | 3.854E-01 |
| F6 | Best | 1.269E-15 | 4.311E+02 | 2.078E-05 | 9.067E-03 | 2.222E-02 | 2.167E+00 | 1.415E-07 | 2.001E+00 | 2.536E-01 | 9.345E-04 | 2.127E-07 |
| | Average | **2.562E-15** | 2.205E+03 | 5.579E-01 | 6.195E-02 | 2.092E+00 | 2.743E+00 | 1.515E-05 | 2.584E+00 | 1.815E+00 | 2.622E-03 | 8.514E-07 |
| | STD | 8.125E-16 | 8.527E+02 | 2.498E-01 | 7.788E-02 | 2.757E+00 | 2.757E-01 | 2.221E-05 | 3.849E-01 | 5.486E-01 | 9.238E-04 | 4.797E-07 |
| F7 | Best | 1.418E-06 | 3.160E-01 | 2.323E-04 | 1.251E-05 | 1.151E-06 | 5.268E-07 | 9.774E-07 | 1.818E-05 | 3.076E-06 | 2.439E-06 | 2.206E-03 |
| | Average | **2.734E-05** | 6.707E-01 | 7.003E-04 | 1.323E-03 | 5.366E-05 | 2.780E-05 | 4.846E-05 | 1.827E-04 | 5.815E-05 | 5.780E-05 | 8.202E-03 |
| | STD | 2.898E-05 | 2.668E-01 | 4.680E-04 | 1.658E-03 | 5.824E-05 | 2.442E-05 | 3.560E-05 | 1.113E-04 | 5.792E-05 | 4.318E-05 | 4.350E-03 |
| F8 | Best | -12569.4866 | -7397.2856 | -8169.8202 | -12569.3327 | -4005.5234 | -7193.5468 | -12567.3149 | -6468.5522 | -8658.7625 | -8522.5366 | -9798.3401 |
| | Average | **-12569.4866** | -5706.4403 | -6359.5788 | -11361.0615 | -2673.1493 | -5599.6796 | -10980.4218 | -4454.0778 | -6900.2174 | -7264.8265 | -8593.5687 |
| | STD | 7.80E-11 | 7.66E+02 | 7.53E+02 | 1.55E+03 | 6.11E+02 | 4.74E+02 | 2.95E+03 | 1.19E+03 | 7.47E+02 | 7.29E+02 | 7.22E+02 |
| F9 | Best | 0.000E+00 | 5.090E+01 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 7.089E-09 |
| | Average | **0.000E+00** | 9.286E+01 | 5.684E-15 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.419E+01 |
| | STD | 0.000E+00 | 2.101E+01 | 1.735E-14 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.024E+01 |
| F10 | Best | 4.441E-16 | 8.181E+00 | 7.550E-15 | 4.441E-16 | 4.441E-16 | 4.441E-16 | 4.441E-16 | 3.997E-15 | 4.441E-16 | 4.441E-16 | 1.023E-05 |
| | Average | **4.441E-16** | 1.077E+01 | 1.454E-14 | 3.168E-15 | **4.441E-16** | **4.441E-16** | **4.441E-16** | 3.997E-15 | **4.441E-16** | **4.441E-16** | 2.754E-05 |
| | STD | 0.000E+00 | 1.171E+00 | 1.741E-15 | 2.224E-15 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 9.596E-06 |
| F11 | Best | 0.000E+00 | 8.809E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.203E-03 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 2.903E-08 |
| | Average | **0.000E+00** | 2.010E+01 | 9.535E-04 | **0.000E+00** | **0.000E+00** | 7.653E-02 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.255E-02 |
| | STD | 0.000E+00 | 6.161E+00 | 5.222E-03 | 0.000E+00 | 0.000E+00 | 6.311E-02 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.943E-02 |
| F12 | Best | 4.994E-17 | 1.248E+01 | 1.378E-02 | 1.212E-03 | 1.343E-04 | 3.198E-01 | 2.265E-09 | 1.057E-01 | 3.900E-02 | 2.259E-05 | 1.580E-08 |
| | Average | **2.581E-16** | 4.949E+02 | 3.357E-02 | 6.138E-03 | 1.932E-04 | 3.939E-01 | 4.327E-07 | 2.411E-01 | 8.888E-02 | 6.084E-05 | 5.829E-08 |
| | STD | 1.168E-16 | 1.450E+03 | 1.434E-02 | 4.340E-03 | 2.330E-05 | 4.386E-02 | 6.276E-07 | 1.086E-01 | 3.982E-02 | 1.851E-05 | 3.062E-08 |
| F13 | Best | 1.942E-15 | 4.828E+02 | 2.543E-05 | 1.381E-02 | 1.385E+00 | 2.569E+00 | 1.415E-08 | 9.927E-01 | 1.126E+00 | 5.361E-04 | 3.084E-07 |
| | Average | **3.684E-15** | 2.632E+05 | 4.446E-01 | 2.086E-01 | 2.894E+00 | 2.770E+00 | 3.014E-06 | 1.581E+00 | 2.281E+00 | 3.012E-01 | 8.914E-07 |
| | STD | 1.224E-15 | 2.825E+05 | 1.813E-01 | 1.810E-01 | 2.882E-01 | 1.195E-01 | 4.439E-06 | 2.327E-01 | 3.994E-01 | 9.061E-01 | 4.568E-07 |
| F14 | Best | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |
| | Average | **0.998** | 5.9745 | 3.2156 | 1.919 | 7.8214 | 8.7481 | 1.7536 | 5.039 | 2.9666 | 10.0884 | **0.998** |
| | STD | 1.09E-16 | 3.15E+00 | 3.8889 | 1.8701 | 3.9601 | 4.2328 | 1.87E+00 | 4.6304 | 2.8093 | 4.2763 | 4.19E-16 |
| F15 | Best | 3.075E-04 | 3.704E-04 | 3.075E-04 | 3.077E-04 | 3.100E-04 | 3.267E-04 | 3.280E-04 | 3.075E-04 | 3.075E-04 | 3.075E-04 | 3.075E-04 |
| | Average | 6.811E-04 | 8.770E-04 | 2.378E-03 | 6.132E-04 | **3.150E-04** | 6.279E-03 | 4.206E-04 | 3.283E-04 | 4.060E-04 | 4.708E-04 | 3.846E-04 |
| | STD | 2.884E-04 | 5.952E-04 | 6.102E-03 | 3.612E-04 | 3.286E-06 | 1.375E-02 | 6.581E-05 | 4.092E-05 | 2.796E-04 | 3.655E-04 | 2.827E-04 |
| F16 | Best | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 |
| | Average | **-1.0316** | **-1.0316** | **-1.0316** | **-1.0316** | -0.91541 | **-1.0316** | -1.0315 | -1.0315 | **-1.0316** | -0.95001 | -1.0315 |
| | STD | 4.70E-16 | 6.52E-16 | 3.36E-09 | 2.62E-11 | 0.21162 | 6.88E-08 | 1.56E-04 | 3.88E-08 | 1.69E-10 | 0.24904 | 1.04E-13 |
| F17 | Best | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39802 | 0.3978 | 0.3978 | 0.39789 | 0.39789 | 0.39789 |
| | Average | **0.39789** | **0.39789** | **0.39789** | **0.39789** | 0.49131 | 0.40582 | 0.39796 | 0.39796 | **0.39789** | **0.39789** | **0.39789** |
| | STD | 0 | 0 | 5.19E-07 | 3.13E-07 | 2.24E-01 | 0.0086516 | 9.14E-05 | 7.72E-06 | 4.12E-08 | 3.58E-11 | 1.37E-12 |
| F18 | Best | 3 | 3 | 3 | 3 | 3.2053 | 3 | 3 | 3 | 3 | 3 | 3 |
| | Average | **3** | 3.9 | **3** | **3** | 17.8425 | 9.977 | 3.0093 | **3** | **3** | 6.6 | **3** |
| | STD | 3.00E-14 | 4.93E+00 | 7.55E-06 | 1.76E-05 | 20.9962 | 11.7881 | 9.67E-03 | 6.08E-07 | 2.08E-06 | 9.3351 | 1.02E-09 |

(*continued on next page*)

**Table 5** (*continued*)

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F19 | Best | -3.8628 | -3.8628 | -3.8628 | -3.8628 | -3.8582 | -3.8602 | -3.8625 | -3.8628 | -3.8628 | -3.8628 | -3.8628 |
| | Average | **-3.8628** | -3.8609 | -3.8625 | -3.8605 | -3.7431 | -3.8539 | -3.8606 | -3.8594 | -3.8605 | **-3.8628** | **-3.8628** |
| | STD | 1.80E-15 | 5.24E-03 | 0.00080988 | 0.0031719 | 1.80E-01 | 0.0028421 | 1.61E-03 | 0.0039545 | 0.0034667 | 3.12E-07 | 1.05E-08 |
| F20 | Best | -3.322 | -3.322 | -3.322 | -3.322 | -3.054 | -3.2397 | -3.3186 | -3.322 | -3.322 | -3.322 | -3.322 |
| | Average | **-3.2943** | -3.2756 | -3.2836 | -3.2631 | -2.7638 | -3.131 | -3.2178 | -3.1473 | -3.1872 | -3.2622 | -3.2824 |
| | STD | 0.051146 | 0.067221 | 0.07351 | 0.079836 | 0.19516 | 0.060747 | 0.085336 | 0.14194 | 0.16423 | 0.060758 | 0.057005 |
| F21 | Best | -10.1532 | -10.1532 | -10.1531 | -10.1531 | -7.0111 | -8.0876 | -10.1531 | -10.1523 | -10.1532 | -10.1532 | -10.1532 |
| | Average | -7.1105 | -6.1463 | -9.9844 | -9.4679 | -4.0099 | -4.1456 | **-10.1511** | -8.9219 | -6.3079 | -5.3966 | -6.8846 |
| | STD | 2.7709 | 3.4464 | 0.92248 | 1.7662 | 1.2772 | 1.4087 | 0.0026246 | 2.2736 | 2.7647 | 1.293 | 3.2357 |
| F22 | Best | -10.4029 | -10.4029 | -10.4029 | -10.4029 | -8.0864 | -8.0937 | -10.4028 | -10.4028 | -10.4029 | -10.4029 | -10.4029 |
| | Average | -7.3261 | -6.6258 | **-10.4026** | -9.4648 | -4.5615 | -4.5638 | -10.4006 | -10.3999 | -7.0393 | -5.9749 | -7.4233 |
| | STD | 2.993 | 3.6585 | 0.00022921 | 2.1272 | 1.5937 | 1.3432 | 0.0026425 | 0.0018579 | 2.6031 | 2.0141 | 3.5816 |
| F23 | Best | -10.5364 | -10.5364 | -10.5363 | -10.5363 | -8.8041 | -8.6799 | -10.5363 | -10.536 | -10.5364 | -10.5364 | -10.5364 |
| | Average | -7.3263 | -6.6696 | **-10.5359** | -9.1397 | -4.347 | -4.2564 | -10.5343 | -10.5331 | -7.1523 | -6.0298 | -7.3603 |
| | STD | 3.1338 | 3.7275 | 0.00020966 | 2.5788 | 1.4485 | 1.7667 | 0.0026796 | 0.0016982 | 2.9099 | 2.0499 | 3.7795 |
| Mean | Rank | 1.96 | 8.61 | 5.48 | 4.87 | 6.61 | 7.83 | 3.78 | 6.04 | 4.87 | 4.78 | 5.35 |
| Final | Ranking | 1 | 11 | 7 | 4 | 9 | 10 | 2 | 8 | 4 | 3 | 6 |

where $F(X_i^t)$ and $F(Prey^{t-t_2})$ are the fitness values of $X_i^t$ and $Prey^{t-t_2}$. $X_i$ is redistributed when $t > t_1$ & $F(X_i^t) = F(Prey^{t-t_2})$, which conducive to escaping from local optimal solutions.

### 2.3.6. Pseudo-code of BSLO

To recap, BSLO first initializes a given number of random candidate solutions. After obtaining an optimal solution by comparing the current optimal solutions with the historical optimal solutions, BSLO updates next positions of candidate solutions using five update strategies, including re-tracking strategy, exploration strategy of directional leeches, exploitation strategy of directional leeches, switching mechanism of directional leeches, and search strategy of directionless leeches. Re-tracking strategy is performed after entering exploration and exploitation for $t_1$ iterations. When leeches are trapped into local solutions for $t_2$ iterations, these leeches search for humans again. The next positions of $N_1$ leeches are obtained using the exploration strategy of directional leeches, the exploitation strategy of directional leeches and the switching mechanism of directional leeches, while the next positions of $N_2$ leeches are updated using the search strategy of directionless leeches. Ultimately, the optimization of BSLO is ended after $t = T$. Algorithm 1 presents the pseudo-code of BSLO.

## 3. Numerical experiments

In this section, BSLO's performance is assessed using three types of test functions, including twenty-three classical benchmark functions, CEC 2017, and CEC 2019 test functions.

### 3.1. Results of BSLO for benchmark functions

Twenty-three classical benchmark functions, a class of commonly used functions to evaluate the exploration and exploitation of SI algorithms, are divided into three groups, including unimodal, multimodal functions, and multimodal test functions with fixed dimension. They are described in Table 2. The exploitation ability and convergence speed of SI algorithms are tested using unimodal functions (F1-F7) with one global optimum. Multimodal functions (F8-F13) with multiple local optima are performed to assess the capabilities of exploration and local optima avoidance of SI algorithms. The third group (F14-F23) is utilized to measure the exploration ability of SI algorithms at low dimension.

#### 3.1.1. Parameter settings of BSLO

Before evaluating the optimization performance of BSLO, the parameters of BSLO need to be fine-tuned to search for better results. BSLO includes six main parameters: $m$ in Eq. (4), $a$ in Eq. (6), $b$ in Eq. (7) for the exploration phase, $b$ in Eq. (7) for the exploitation phase ($b_2$), $t_1$ in Eq. (20), and $t_2$ in Eq. (20). As described in Section 2, $m$ represents the directional leeches, which are in the majority, therefore, $m$ takes values in the range [0.7, 1]. Extensive preliminary experiments using the following parameter setting method described in the following part were performed to confirm the approximate ranges of other parameters ($a$, $b$, $b_2$, $t_1$, $t_2$). After preliminary experiments, six parameters are divided into three levels, as shown in Table 3. As expected, $a$ is very closed to 1, $b$ is a very small value, $b_2$ is smaller value than $b$ for accurately searching for the optimal solution in exploitation phase. $t_1$ and $t_2$ are small values since some leeches can find human at the beginning, and $t_1$ is bigger than or equal to $t_2$. The result obtained by the method of permutations and combinations, subtracting the scenario where $t_1$ is bigger than or equal to $t_2$, yields a total of 486 parameter settings. Then, these parameter settings are used to optimize twenty-three classical benchmark functions with 30 dimensions. Here, the number of function evaluations $FES = 30000$, the population size $N = 30$ and the maximum iteration $T = 1000$ are used to search for the optimal solution for the benchmark functions, and BSLO is run for 30 times for each function to ensure the experimental reliability. A statistical method, called Friedman ranking test, is

**Table 6**
Wilcoxon rank sum test between BSLO and other ten algorithms for the 23 classical test functions.

| F | PSO p (Sig.) | GWO p (Sig.) | WOA p (Sig.) | SHO p (Sig.) | AOA p (Sig.) | AO p (Sig.) | GJO p (Sig.) | SCSO p (Sig.) | FOX p (Sig.) | DO p (Sig.) |
|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) |
| F2 | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | NaN (~) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) |
| F3 | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) |
| F4 | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | NaN (~) | 1.21E-12 (+) |
| F5 | 3.02E-11 (+) | 9.76E-10 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (-) | 3.02E-11 (-) | 3.02E-11 (+) | 3.02E-11 (+) | 9.53E-07 (+) |
| F6 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F7 | 3.02E-11 (+) | 4.08E-11 (+) | 1.96E-10 (+) | 4.12E-01 (~) | 8.77E-02 (~) | 2.89E-03 (+) | 4.31E-08 (+) | 2.50E-03 (+) | 5.08E-03 (+) | 3.02E-11 (+) |
| F8 | 3.01E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F9 | 1.21E-12 (+) | 6.52E-04 (+) | NaN (~) | NaN (~) | NaN (~) | NaN (~) | NaN (~) | NaN (~) | NaN (~) | 1.21E-12 (+) |
| F10 | 1.21E-12 (+) | 4.47E-13 (+) | 2.53E-11 (+) | NaN (~) | NaN (~) | NaN (~) | 1.18E-13 (+) | NaN (~) | NaN (~) | 1.21E-12 (+) |
| F11 | 1.21E-12 (+) | 4.19E-02 (+) | 4.19E-02 (+) | NaN (~) | 1.21E-12 (+) | NaN (~) | NaN (~) | NaN (~) | NaN (~) | 1.21E-12 (+) |
| F12 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F13 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F14 | 1.19E-09 (+) | 1.46E-11 (+) | 1.46E-11 (+) | 1.44E-11 (+) | 1.46E-11 (+) | 1.46E-11 (+) | 1.46E-11 (+) | 1.46E-11 (+) | 1.44E-11 (+) | 1.16E-11 (+) |
| F15 | 2.53E-04 (+) | 1.38E-02 (+) | 8.07E-01 (~) | 9.51E-06 (-) | 1.45E-01 (~) | 6.67E-03 (-) | 1.53E-05 (-) | 1.25E-04 (-) | 6.38E-03 (-) | 4.44E-07 (-) |
| F16 | 1.32E-10 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.14E-12 (+) | 5.13E-12 (+) |
| F17 | NaN (~) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 1.21E-12 (+) | 4.57E-12 (+) |
| F18 | 6.86E-06 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 3.01E-11 (+) | 4.06E-11 (+) |
| F19 | 6.35E-03 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) | 2.03E-11 (+) |
| F20 | 3.45E-02 (+) | 3.85E-04 (+) | 3.85E-04 (+) | 2.67E-11 (+) | 2.67E-11 (+) | 1.72E-03 (+) | 1.88E-07 (+) | 3.85E-04 (+) | 3.85E-04 (+) | 2.12E-02 (+) |
| F21 | 4.30E-02 (+) | 4.81E-02 (-) | 1.33E-01 (~) | 4.93E-06 (+) | 1.49E-05 (+) | 1.92E-03 (-) | 3.12E-02 (-) | 7.56E-04 (+) | 1.74E-03 (+) | 5.79E-01(~) |
| F22 | 1.80E-02 (+) | 7.68E-02 (~) | 6.84E-01 (~) | 1.16E-02 (+) | 2.69E-02 (+) | 7.68E-02 (~) | 1.05E-01(~) | 3.47E-01(~) | 1.75E-01(~) | 6.95E-01(~) |
| F23 | 3.31E-02 (+) | 1.85E-01 (~) | 3.63E-01 (~) | 1.16E-02 (+) | 6.04E-03 (+) | 1.85E-01 (~) | 4.28E-01(~) | 1.53E-01(~) | 2.24E-03 | 1.49E-01(~) |
| (W\|L\|T) | (22\|0\|1) | (20\|1\|2) | (18\|0\|5) | (14\|1\|8) | (18\|0\|5) | (15\|3\|5) | (17\|2\|4) | (17\|1\|5) | (14\|1\|8) | (19\|1\|3) |

utilized to estimate the optimization results of these parameter settings. The results of the top thirty parameter settings are shown in Fig. 4. Results show that BSLO with parameter settings ($m, a, b, b_2, t_1, t_2$) = (0.8, 0.97, 0.001, 0.00001, 20, 20) obtains the optimal performance. Also, it is obvious that BSLO always gets a good ranking when the proportion of directional leeches at the beginning is 80%. Therefore, BSLO with the parameter setting ($m, a, b, b_2, t_1, t_2$) = (0.8, 0.97, 0.001, 0.00001, 20, 20) is used in this paper.

### 3.1.2. Qualitive analysis of BSLO

To qualitatively analyze the optimization process and convergence behaviors of BSLO, four common criteria are performed, and the results are shown in Fig. 5. Four criteria include the search history of candidate solutions, the 1D trajectory of the first candidate solution, the average fitness values of candidate solutions, and the convergence curve of BSLO, which are given in Fig. 5 in the second, third, fourth, and fifth column, respectively. The 2D graph of twenty-three benchmark functions to observe their topology is shown in the first column.

Search history is commonly used for studying the collective and interactive behaviors between candidate solutions and the optimal solution, which is an intuitive way to show the modality of candidate solutions. For the unimodal functions, the modality of candidate solutions shows the collective behavior, which means that candidate solutions search for the space surrounding the optimal solution. As shown in the search history of BSLO for F1-F7, candidate solutions can search for the whole space and always cluster around the optimal solution, which indicates that BSLO possesses good exploitation around the optimal solution. In contrast, the search history for multimodal functions with many obvious local optimal solutions can show a scattering, as shown in the search history of BSLO for F8 and F14, which is because multimodal functions, with local optima, are more intricate than unimodal functions with only a global optimum. Hence, the search history is applied to enhance the exploration and exploitation performance of BSLO.

The trajectory curve indicates that the trajectory of leeches has a large amplitude, but soon approaches 0, which demonstrates that BSLO owns strong exploration and exploitation performance and high convergence speed. It can be noted that for solving multimodal functions, BSLO has many large amplitude fluctuations during the iterations, which is caused by the existence of local optima. Hence, candidate solutions are easier to be fallen into local optima for multimodal functions, but the re-tracking strategy of BSLO can make candidate solutions jump out local optima and look for the optimal solution again, which increases the BSLO's ability of local optima avoidance.

The average fitness curve shows that candidate solutions exhibit high average fitness values at the beginning, but these values soon reduce to a very low value, which also indicates that BSLO possesses a very fast convergence speed. Also, the average fitness value of BSLO for most functions presents a drop in volatility in the later iterations, which is the result of candidate solutions constantly exploring and approaching the optimal solution. Similarly, the convergence curve also shows that BSLO owns a very fast convergence speed and can approach the optimal solution after a few iterations. In short, four qualitative experiments demonstrate that BSLO owns a nice balance strategy between exploration and exploitation and has a fast convergence speed.

### 3.1.3. Results comparisons between BSLO and other algorithms

The optimization performance of BSLO is further analyzed using the 23 benchmar functions with 30 dimensions. Several comparative algorithms are used: six well-known algorithms and four new algorithms, including Particle Swarm Optimization (PSO) [94], Grey Wolf Optimizer (GWO) [95], Whale Optimization Algorithm (WOA) [96], Spotted Hyena Optimizer (SHO) [97], Arithmetic Optimization Algorithm (AOA) [98], Aquila Optimizer (AO) [99], Golden Jackal Optimization (GJO) [100], Sand Cat Swarm Optimization (SCSO) [101], Fox optimizer (FOX) [102], and Dandelion Optimizer (DO) [103]. These comparative algorithms have been evaluated by the 23 benchmar functions or CEC benchmark functions in the original publication, which guarantees the experimental effectiveness. The important parameters of BSLO and comparative algorithms are described in Table 4. The maximum iteration and number of candidate solutions are fixed as 1000 and 30. Hence, each algorithm executes 30000 fitness evaluations for each benchmark function, and is performed 30 times independently to guarantee experimental reliability.

The optimization results are presented using the best fitness value (Best), average fitness values (Average) and standard deviation (STD). In addition, Friedman ranking test and Wilcoxon signed-rank test, two statistical methods, are used to evaluate the performance of BSLO and comparative algorithms. Results of three criteria and the Friedman

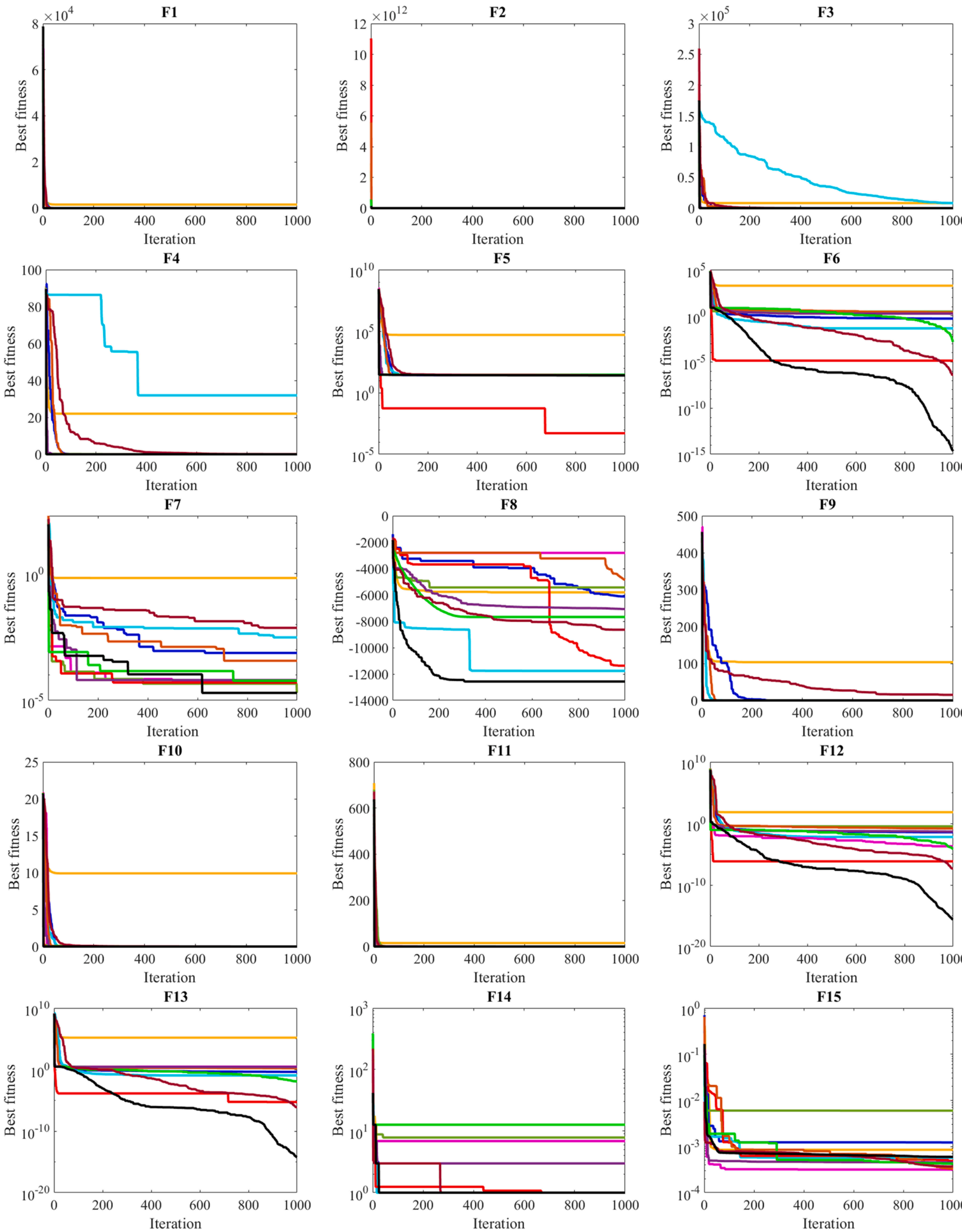


**Fig. 6.** Convergence curves of BSLO and ten algorithms for twenty-three benchmark functions.

ranking test are presented in Table 5. It is evident that BSLO gets eighteen best average optimal solutions and seventeen best STD compared to other algorithms, accounting for 78% and 74%, respectively. For the unimodal functions (F1-F7), BSLO gets six best average optimal solutions and one suboptimal solution, ranked the first, which indicates that BSLO owns stronger exploitation ability than comparative algorithms. It can be obviously observed that BSLO gets all optimal solutions of the multimodal functions (F8-F13) compared with other algorithms, ranked the first, which shows that BSLO has strong exploration ability and easier to escape from local optimal solutions. Furthermore, Friedman

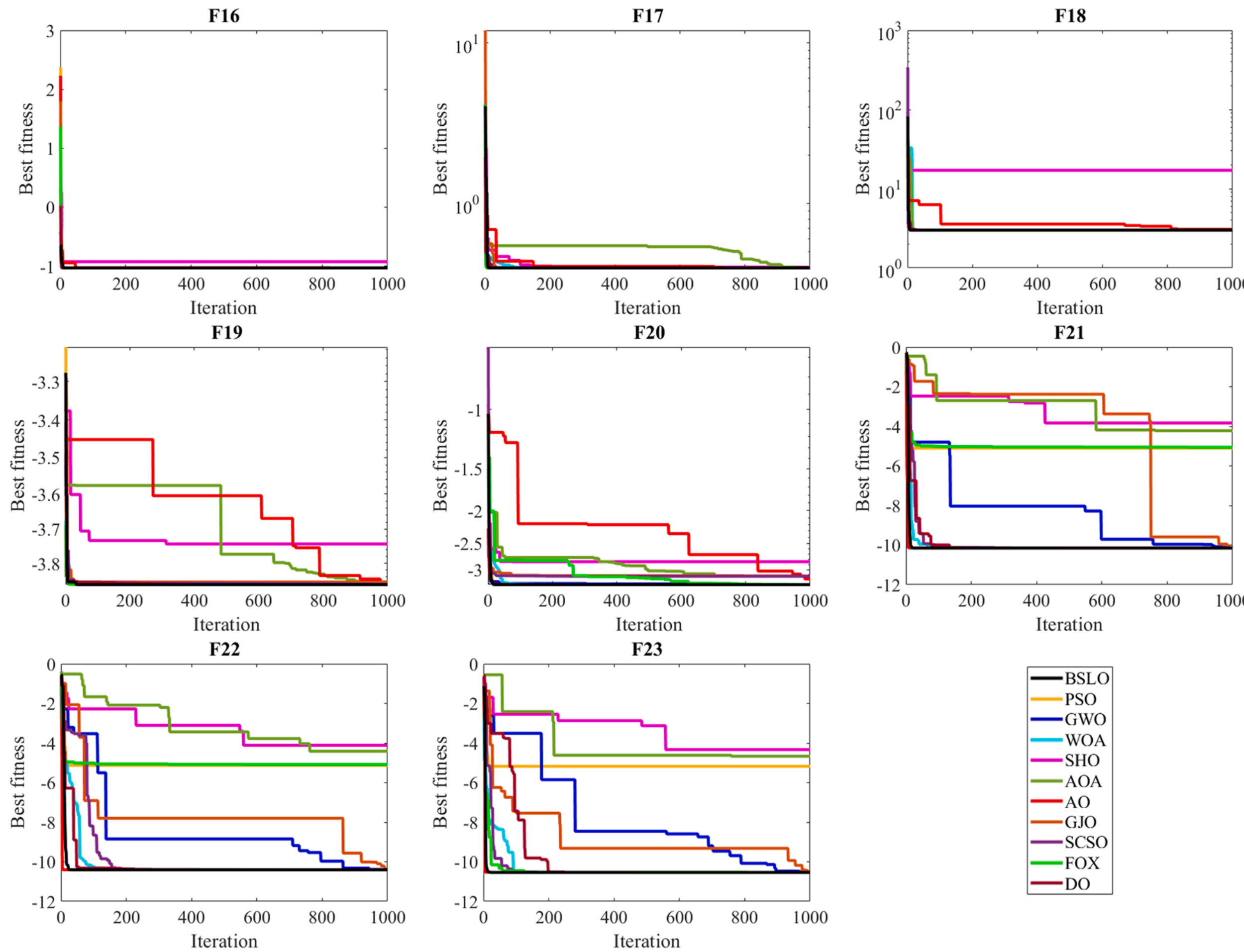


**Fig. 6.** (*continued*).

ranking test shows that BSLO obtains the lowest mean values, ranked the first, followed by AO, FOX, SCSO, WOA, DO, GWO, SHO, AOA, PSO.

The Wilcoxon rank sum test given in Table 6 presents the significant differences between BSLO and ten algorithms for these functions. $\alpha$ represents a significant level and is equal to 0.05. When $\alpha > 0.05$, there is no significant difference between BSLO and comparative algorithms, and results are signed by "$\sim$". Otherwise, BSLO has significant differences in comparison to other algorithms, and "+" denotes that BSLO wins other algorithms and "-" shows that BSLO loses. Table 6 indicates that BSLO wins against other comparative algorithms, especially for PSO, GWO, WOA, AOA, GJO, SCSO, DO.

To intuitively observe the optimization process, the convergence curves of BSLO and comparative algorithms are plotted for these test functions and shown in Fig. 6. It is evident that BSLO possesses very fast convergence speed and can converge to a small value with a small number of iterations for almost all functions except for F15, among which BSLO has fastest convergence speed for F1-F4, F9-F11, F18, and F20. Results also show that BSLO owns slower convergence speed than AO for F6, F12, and F13 at the beginning, but BSLO soon converges to smaller values. In addition, unlike most other algorithms, BSLO always significantly converges to smaller values for F6, F12 and F13, which illustrates that BSLO has a strong ability of local optima avoidance. Therefore, BSLO has the best convergence speed compared with ten algorithms for twenty-three benchmark functions.

### 3.1.4. Stability analysis of BSLO

To evaluate its stability, BSLO and ten other algorithms are tested for unimodal and multimodal functions of 23 benchmark functions with different dimensions. The parameter settings of all algorithms are the same as before except for dimensions. The average fitness value, STD, and Friedman ranking test are selected to evaluate BSLO and other algorithms, as shown in Table 7. Friedman ranking test results show that BSLO is ranked first in all dimensions for these functions. It can be observed that BSLO wins other algorithms in terms of the number of average fitness values. In addition, BSLO is also ranked first in the quantity of STD for these functions with different dimensions. Although BSLO shows optimal results for these functions with $D = 30$, $D = 100$ and $D = 500$, the results of $D = 30$ and $D = 100$ indicate that BSLO possesses more outstanding optimization abilities for solving problems with low and medium dimensions. Overall, BSLO has good stability for different dimensions.

## 3.2. Results of BSLO for CEC 2017 test functions

To study the optimization performance of BSLO, more complex test functions are selected, namely CEC 2017 test functions. CEC 2017 with 29 functions includes four types of functions, including unimodal functions (F1 and F3), simple multimodal functions (F4-F10), hybrid functions (F11-F20), and composite functions (F21-F30). The first two functions are performed to assess BSLO's optimization ability, while more complex hybrid and composite functions are applied to test BSLO's ability of local optima avoidance. The detail information of CEC 2017 can be found in Ref. [99]. The comparative algorithms are still PSO, GWO, WOA, SHO, AOA, AO, GJO, SCSO, FOX, and DO. The maximum iterations, number of candidate solutions and initialization of other parameters remain the same as before except for the dimension fixed as 10. Each algorithm is independently performed 30 times for every function.

**Table 7**
Results of BSLO and other ten algorithms for thirteen classical test functions (F1-F13) with different dimensions $D$=30, 100, and 500 (the best average optimal solution is marked in bold).

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1_30D | Average | **0.000E+00** | **2.203E+03** | 4.736E-59 | 1.782E-153 | **0.000E+00** | 7.633E-80 | 8.940E-219 | 4.532E-113 | 1.004E-228 | **0.000E+00** | 1.587E-08 |
| | STD | 0.000E+00 | 6.062E+02 | 8.670E-59 | 4.777E-153 | 0.000E+00 | 4.167E-79 | 0.000E+00 | 1.398E-112 | 0.000E+00 | 0.000E+00 | 1.113E-08 |
| | Rank | 1 | 11 | 9 | 6 | 1 | 8 | 5 | 7 | 4 | 1 | 10 |
| F1_100D | Average | **0.000E+00** | 2.583E+04 | 1.749E-29 | 6.672E-149 | **0.000E+00** | 1.914E-02 | 2.824E-203 | 1.149E-59 | 5.654E-215 | **0.000E+00** | 9.579E-03 |
| | STD | 0.000E+00 | 3.356E+03 | 2.227E-29 | 2.347E-148 | 0.000E+00 | 8.911E-03 | 0.000E+00 | 2.961E-59 | 0.000E+00 | 0.000E+00 | 2.937E-03 |
| | Rank | 1 | 11 | 8 | 6 | 1 | 10 | 5 | 7 | 4 | 1 | 9 |
| F1_500D | Average | **0.000E+00** | 2.317E+05 | 1.471E-12 | 4.508E-145 | **0.000E+00** | 5.985E-01 | 1.677E-204 | 1.884E-32 | 2.554E-205 | **0.000E+00** | 6.354E+02 |
| | STD | 0.000E+00 | 1.197E+04 | 9.453E-13 | 2.294E-144 | 0.000E+00 | 4.141E-02 | 0.000E+00 | 1.668E-32 | 0.000E+00 | 0.000E+00 | 1.020E+02 |
| | Rank | 1 | 11 | 8 | 6 | 1 | 9 | 5 | 7 | 4 | 1 | 10 |
| F2_30D | Average | **0.000E+00** | 2.329E+01 | 6.669E-35 | 1.980E-104 | **0.000E+00** | **0.000E+00** | 1.853E-106 | 1.819E-66 | 6.623E-121 | **0.000E+00** | 8.490E-05 |
| | STD | 0.000E+00 | 6.519E+00 | 6.871E-35 | 9.051E-104 | 0.000E+00 | 0.000E+00 | 1.015E-105 | 2.792E-66 | 3.578E-120 | 0.000E+00 | 6.327E-05 |
| | Rank | 1 | 11 | 9 | 7 | 1 | 1 | 6 | 8 | 5 | 1 | 10 |
| F2_100D | Average | **0.000E+00** | 1.322E+02 | 5.430E-18 | 1.483E-101 | **0.000E+00** | 8.495E-130 | 4.050E-106 | 2.472E-37 | 3.993E-114 | **0.000E+00** | 6.140E-02 |
| | STD | 0.000E+00 | 1.306E+01 | 3.057E-18 | 8.072E-101 | 0.000E+00 | 4.396E-129 | 2.218E-105 | 2.469E-37 | 1.219E-113 | 0.000E+00 | 1.285E-02 |
| | Rank | 1 | 11 | 9 | 7 | 1 | 4 | 6 | 8 | 5 | 1 | 10 |
| F2_500D | Average | **0.000E+00** | 8.958E+02 | 6.243E-08 | 2.829E-101 | **0.000E+00** | 1.662E-04 | 1.859E-106 | 8.936E-21 | 2.739E-107 | **0.000E+00** | 2.006E+01 |
| | STD | 0.000E+00 | 6.530E+01 | 1.610E-08 | 7.080E-101 | 0.000E+00 | 3.060E-04 | 0.000E+00 | 5.375E-21 | 7.427E-107 | 0.000E+00 | 1.363E+00 |
| | Rank | 1 | 11 | 8 | 6 | 1 | 9 | 5 | 7 | 4 | 1 | 10 |
| F3_30D | Average | **0.000E+00** | **8.146E+03** | 1.064E-15 | 1.902E+04 | **0.000E+00** | 5.994E-04 | 1.558E-201 | 3.041E-38 | 2.753E-204 | **0.000E+00** | 5.707E-01 |
| | STD | 0.000E+00 | 3.081E+03 | 2.979E-15 | 1.072E+04 | 0.000E+00 | 1.973E-03 | 0.000E+00 | 1.613E-37 | 0.000E+00 | 0.000E+00 | 4.194E-01 |
| | Rank | 1 | 10 | 7 | 11 | 1 | 8 | 5 | 6 | 4 | 1 | 9 |
| F3_100D | Average | **0.000E+00** | 1.013E+05 | 5.452E+00 | 8.772E+05 | **0.000E+00** | 5.997E-01 | 4.735E-196 | 5.586E-07 | 2.367E-188 | **0.000E+00** | 1.039E+04 |
| | STD | 0.000E+00 | 3.722E+04 | 8.789E+00 | 1.684E+05 | 0.000E+00 | 5.464E-01 | 0.000E+00 | 3.032E-06 | 0.000E+00 | 0.000E+00 | 4.170E+03 |
| | Rank | 1 | 10 | 8 | 11 | 1 | 7 | 4 | 6 | 5 | 1 | 9 |
| F3_500D | Average | **0.000E+00** | 2.559E+06 | 1.328E+05 | 2.812E+07 | **0.000E+00** | 2.888E+01 | 2.623E-202 | 5.529E+03 | 3.310E-181 | **0.000E+00** | 1.066E+06 |
| | STD | 0.000E+00 | 7.909E+05 | 5.336E+04 | 9.789E+06 | 0.000E+00 | 1.620E+01 | 0.000E+00 | 1.347E+04 | 0.000E+00 | 0.000E+00 | 1.935E+05 |
| | Rank | 1 | 10 | 8 | 11 | 1 | 6 | 4 | 7 | 5 | 1 | 9 |
| F4_30D | Average | **0.000E+00** | **2.277E+01** | 1.369E-14 | 3.111E+01 | **0.000E+00** | 1.579E-02 | 1.321E-117 | 2.823E-33 | 6.399E-100 | **0.000E+00** | 7.577E-02 |
| | STD | 0.000E+00 | 4.038E+00 | 2.117E-14 | 3.218E+01 | 0.000E+00 | 2.078E-02 | 7.038E-117 | 8.144E-33 | 3.471E-99 | 0.000E+00 | 4.736E-02 |
| | Rank | 1 | 10 | 7 | 11 | 1 | 8 | 4 | 6 | 5 | 1 | 9 |
| F4_100D | Average | **0.000E+00** | 4.000E+01 | 4.558E-03 | 6.417E+01 | **0.000E+00** | 8.662E-02 | 4.982E-105 | 1.148E+00 | 3.257E-97 | **0.000E+00** | 4.581E+01 |
| | STD | 0.000E+00 | 3.704E+00 | 1.070E-02 | 2.765E+01 | 0.000E+00 | 1.147E-02 | 2.729E-104 | 4.371E+00 | 1.315E-96 | 0.000E+00 | 9.369E+00 |
| | Rank | 1 | 9 | 6 | 11 | 1 | 7 | 4 | 8 | 5 | 1 | 10 |
| F4_500D | Average | **0.000E+00** | 5.777E+01 | 5.640E+01 | 7.881E+01 | **0.000E+00** | 1.682E-01 | 5.311E-103 | 7.684E+01 | 1.640E-94 | **0.000E+00** | 9.299E+01 |
| | STD | 0.000E+00 | 2.796E+00 | 5.663E+00 | 2.183E+01 | 0.000E+00 | 1.151E-02 | 0.000E+00 | 5.482E+00 | 5.251E-94 | 0.000E+00 | 1.533E+00 |
| | Rank | 1 | 8 | 7 | 10 | 1 | 6 | 4 | 9 | 5 | 1 | 11 |
| F5_30D | Average | 2.522E+01 | 4.357E+05 | 2.663E+01 | 2.695E+01 | 2.880E+01 | 2.810E+01 | **7.451E-04** | 2.754E+01 | 2.805E+01 | 2.876E+01 | 2.598E+01 |
| | STD | 1.337E-01 | 3.755E+05 | 9.130E-01 | 3.426E-01 | 1.173E-01 | 4.831E-01 | 7.925E-04 | 8.071E-01 | 9.879E-01 | 3.486E-02 | 3.854E-01 |
| | Rank | 2 | 11 | 4 | 5 | 10 | 8 | 1 | 6 | 7 | 9 | 3 |
| F5_100D | Average | 9.532E+01 | 1.545E+07 | 9.751E+01 | 9.768E+01 | 9.882E+01 | 9.884E+01 | **4.971E-03** | 9.817E+01 | 9.831E+01 | 9.839E+01 | 1.110E+02 |
| | STD | 1.308E-01 | 4.256E+06 | 7.622E-01 | 4.277E-01 | 3.078E-01 | 1.361E-01 | 1.086E-02 | 6.303E-01 | 5.094E-01 | 1.993E-01 | 3.011E+01 |
| | Rank | 2 | 11 | 3 | 4 | 8 | 9 | 1 | 5 | 6 | 7 | 10 |
| F5_500D | Average | 4.941E+02 | 2.437E+08 | 4.976E+02 | 4.957E+02 | 4.989E+02 | 4.991E+02 | **5.085E-02** | 4.984E+02 | 4.894E+02 | 4.964E+02 | 3.993E+05 |
| | STD | 3.121E-01 | 2.871E+07 | 3.718E-01 | 3.383E-01 | 1.436E-01 | 7.001E-02 | 8.834E-02 | 1.145E-01 | 2.913E-01 | 1.553E+00 | 2.324E+05 |
| | Rank | 3 | 11 | 6 | 4 | 7 | 9 | 1 | 8 | 2 | 5 | 10 |
| F6_30D | Average | **2.562E-15** | 2.205E+03 | 5.579E-01 | 6.195E-02 | 2.092E+00 | 2.743E+00 | 1.515E-05 | 2.584E+00 | 1.815E+00 | 2.622E-03 | 8.514E-07 |
| | STD | 8.125E-16 | 8.527E+02 | 2.498E-01 | 7.788E-02 | 2.757E+00 | 2.757E-01 | 2.221E-05 | 3.849E-01 | 5.486E-01 | 9.238E-04 | 4.797E-07 |
| | Rank | 1 | 11 | 6 | 5 | 8 | 10 | 3 | 9 | 7 | 4 | 2 |
| F6_100D | Average | **1.925E-06** | 2.586E+04 | 9.211E+00 | 1.789E+00 | 1.996E+01 | 1.744E+01 | 1.424E-04 | 1.674E+01 | 1.306E+01 | 5.395E-01 | 4.932E-03 |
| | STD | 2.070E-06 | 3.510E+03 | 9.558E-01 | 5.609E-01 | 4.844E+00 | 6.304E-01 | 2.122E-04 | 6.957E-01 | 1.576E+00 | 7.210E-02 | 1.225E-03 |
| | Rank | 1 | 11 | 6 | 5 | 10 | 9 | 2 | 8 | 7 | 4 | 3 |
| F6_500D | Average | 1.921E+01 | 2.255E+05 | 9.364E+01 | 2.002E+01 | 1.197E+02 | 1.143E+02 | **2.498E-04** | 1.118E+02 | 1.025E+02 | 4.808E+01 | 7.413E+02 |
| | STD | 1.939E+00 | 1.214E+04 | 1.643E+00 | 4.833E+00 | 6.853E+00 | 1.263E+00 | 3.382E-04 | 1.504E+00 | 4.237E+00 | 1.983E+01 | 1.958E+02 |
| | Rank | 2 | 11 | 5 | 3 | 9 | 8 | 1 | 7 | 6 | 4 | 10 |

(*continued on next page*)

**Table 7** (*continued*)

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F7_30D | Average | **2.734E-05** | 6.707E-01 | 7.003E-04 | 1.323E-03 | 5.366E-05 | 2.780E-05 | 4.846E-05 | 1.827E-04 | 5.815E-05 | 5.780E-05 | 8.202E-03 |
| | STD | 2.898E-05 | 2.668E-01 | 4.680E-04 | 1.658E-03 | 5.824E-05 | 2.442E-05 | 3.560E-05 | 1.113E-04 | 5.792E-05 | 4.318E-05 | 4.350E-03 |
| | Rank | 1 | 11 | 8 | 9 | 4 | 2 | 3 | 7 | 6 | 5 | 10 |
| F7_100D | Average | **2.530E-05** | 2.860E+01 | 2.491E-03 | 1.209E-03 | 6.575E-05 | 3.653E-05 | 4.698E-05 | 5.483E-04 | 1.082E-04 | 5.492E-05 | 1.119E-01 |
| | STD | 2.887E-05 | 6.241E+00 | 9.135E-04 | 1.795E-03 | 1.103E-04 | 3.674E-05 | 5.093E-05 | 4.011E-04 | 1.697E-04 | 6.187E-05 | 3.106E-02 |
| | Rank | 1 | 11 | 9 | 8 | 5 | 2 | 3 | 7 | 6 | 4 | 10 |
| F7_500D | Average | **3.315E-05** | 2.212E+03 | 1.063E-02 | 2.182E-03 | 4.226E-05 | 4.258E-05 | 5.369E-05 | 1.222E-03 | 1.239E-04 | 5.871E-05 | 9.983E+00 |
| | STD | 3.453E-05 | 2.377E+02 | 3.269E-03 | 2.263E-03 | 6.318E-05 | 3.190E-05 | 6.023E-05 | 7.061E-04 | 1.397E-04 | 7.583E-05 | 1.510E+00 |
| | Rank | 1 | 11 | 9 | 8 | 2 | 3 | 4 | 7 | 6 | 5 | 10 |
| F8_30D | Average | **-1.257E+04** | -5.706E+03 | -6.360E+03 | -1.136E+04 | -2.673E+03 | -5.600E+03 | -1.098E+04 | -4.454E+03 | -6.900E+03 | -7.265E+03 | -8.594E+03 |
| | STD | 7.796E-11 | 7.663E+02 | 7.534E+02 | 1.549E+03 | 6.114E+02 | 4.738E+02 | 2.951E+03 | 1.190E+03 | 7.473E+02 | 7.292E+02 | 7.223E+02 |
| | Rank | 1 | 8 | 7 | 2 | 11 | 9 | 3 | 10 | 6 | 5 | 4 |
| F8_100D | Average | -3.334E+04 | -1.353E+04 | -1.686E+04 | **-3.836E+04** | -4.862E+03 | -1.072E+04 | -1.261E+04 | -1.043E+04 | -2.033E+04 | -2.295E+04 | -2.198E+04 |
| | STD | 1.405E+03 | 1.676E+03 | 1.247E+03 | 4.758E+03 | 1.292E+03 | 6.789E+02 | 5.672E+03 | 4.214E+03 | 1.773E+03 | 1.016E+03 | 1.534E+03 |
| | Rank | 2 | 7 | 6 | 1 | 11 | 9 | 8 | 10 | 5 | 3 | 4 |
| F8_500D | Average | -9.144E+04 | -3.210E+04 | -6.091E+04 | **-1.933E+05** | -1.289E+04 | -2.459E+04 | -4.745E+04 | -3.381E+04 | -6.963E+04 | -8.455E+04 | -8.142E+04 |
| | STD | 2.310E+03 | 3.664E+03 | 4.269E+03 | 2.285E+04 | 4.099E+03 | 1.581E+03 | 1.669E+04 | 1.746E+04 | 5.563E+03 | 2.041E+03 | 6.500E+03 |
| | Rank | 2 | 9 | 6 | 1 | 11 | 10 | 7 | 8 | 5 | 3 | 4 |
| F9_30D | Average | **0.000E+00** | 9.286E+01 | 5.684E-15 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.419E+01 |
| | STD | 0.000E+00 | 2.101E+01 | 1.735E-14 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.024E+01 |
| | Rank | 1 | 11 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 |
| F9_100D | Average | **0.000E+00** | 6.348E+02 | 2.941E-01 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.118E+02 |
| | STD | 0.000E+00 | 3.643E+01 | 1.093E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 7.552E+01 |
| | Rank | 1 | 11 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 |
| F9_500D | Average | **0.000E+00** | 4.652E+03 | 3.907E+00 | **0.000E+00** | **0.000E+00** | 1.615E-06 | **0.000E+00** | 3.638E-13 | **0.000E+00** | **0.000E+00** | 3.502E+00 |
| | STD | 0.000E+00 | 1.339E+02 | 4.976E+00 | 0.000E+00 | 0.000E+00 | 2.529E-06 | 0.000E+00 | 5.652E-13 | 0.000E+00 | 0.000E+00 | 2.651E-01 |
| | Rank | 1 | 11 | 10 | 1 | 1 | 8 | 1 | 7 | 1 | 1 | 9 |
| F10_30D | Average | **4.441E-16** | 1.077E+01 | 1.454E-14 | 3.168E-15 | **4.441E-16** | **4.441E-16** | **4.441E-16** | 3.997E-15 | **4.441E-16** | **4.441E-16** | 2.754E-05 |
| | STD | 0.000E+00 | 1.171E+00 | 1.741E-15 | 2.224E-15 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 9.596E-06 |
| | Rank | 1 | 11 | 9 | 7 | 1 | 1 | 1 | 8 | 1 | 1 | 10 |
| F10_100D | Average | **4.441E-16** | 1.462E+01 | 1.108E-13 | 3.523E-15 | 6.459E-01 | 7.728E-05 | **4.441E-16** | 8.734E-15 | **4.441E-16** | **4.441E-16** | 1.334E-02 |
| | STD | 0.000E+00 | 4.762E-01 | 1.316E-14 | 2.234E-15 | 2.631E+00 | 4.233E-04 | 0.000E+00 | 2.693E-15 | 0.000E+00 | 0.000E+00 | 4.635E-03 |
| | Rank | 1 | 11 | 7 | 5 | 10 | 8 | 1 | 6 | 1 | 1 | 9 |
| F10_500D | Average | **4.441E-16** | 1.663E+01 | 5.482E-08 | 3.523E-15 | 2.721E+00 | 7.561E-03 | **4.441E-16** | 3.739E-14 | **4.441E-16** | **4.441E-16** | 7.019E+00 |
| | STD | 0.000E+00 | 1.871E-01 | 1.220E-08 | 2.757E-15 | 2.815E+00 | 3.756E-04 | 0.000E+00 | 4.024E-15 | 0.000E+00 | 0.000E+00 | 1.007E+00 |
| | Rank | 1 | 11 | 7 | 5 | 9 | 8 | 1 | 6 | 1 | 1 | 10 |
| F11_30D | Average | **0.000E+00** | 2.010E+01 | 9.535E-04 | **0.000E+00** | **0.000E+00** | 7.653E-02 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.255E-02 |
| | STD | 0.000E+00 | 6.161E+00 | 5.222E-03 | 0.000E+00 | 0.000E+00 | 6.311E-02 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.943E-02 |
| | Rank | 1 | 11 | 8 | 1 | 1 | 11 | 1 | 1 | 1 | 1 | 9 |
| F11_100D | Average | **0.000E+00** | 2.273E+02 | 4.348E-04 | **0.000E+00** | **0.000E+00** | 3.573E+02 | **0.000E+00** | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.474E-02 |
| | STD | 0.000E+00 | 2.811E+01 | 2.382E-03 | 0.000E+00 | 0.000E+00 | 1.179E+02 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 7.358E-03 |
| | Rank | 1 | 10 | 8 | 1 | 1 | 11 | 1 | 1 | 1 | 1 | 9 |
| F11_500D | Average | **0.000E+00** | 2.061E+03 | 2.381E-03 | **0.000E+00** | **0.000E+00** | 7.889E+03 | **0.000E+00** | 7.402E-17 | **0.000E+00** | **0.000E+00** | 7.109E+00 |
| | STD | 0.000E+00 | 1.182E+02 | 9.067E-03 | 0.000E+00 | 0.000E+00 | 1.647E+03 | 0.000E+00 | 5.323E-17 | 0.000E+00 | 0.000E+00 | 9.535E-01 |
| | Rank | 1 | 10 | 8 | 1 | 1 | 11 | 1 | 7 | 1 | 1 | 9 |
| F12_30D | Average | **2.581E-16** | 4.949E+02 | 3.357E-02 | 6.138E-03 | 1.932E-04 | 3.939E-01 | 4.327E-07 | 2.411E-01 | 8.888E-02 | 6.084E-05 | 5.829E-08 |
| | STD | 1.168E-16 | 1.450E+03 | 1.434E-02 | 4.340E-03 | 2.330E-05 | 4.386E-02 | 6.276E-07 | 1.086E-01 | 3.982E-02 | 1.851E-05 | 3.062E-08 |
| | Rank | 1 | 11 | 8 | 6 | 5 | 11 | 3 | 10 | 9 | 4 | 2 |
| F12_100D | Average | **1.697E-08** | 1.881E+06 | 2.389E-01 | 1.570E-02 | 2.737E-04 | 8.606E-01 | 3.051E-07 | 6.076E-01 | 3.117E-01 | 2.143E-02 | 1.565E+00 |
| | STD | 1.422E-08 | 1.229E+06 | 4.493E-02 | 7.198E-03 | 1.300E-05 | 2.479E-02 | 3.986E-07 | 5.628E-02 | 7.815E-02 | 2.721E-02 | 1.012E+00 |
| | Rank | 1 | 11 | 6 | 4 | 3 | 9 | 2 | 8 | 7 | 5 | 10 |
| F12_500D | Average | 6.838E-02 | 1.673E+08 | 7.432E-01 | 4.118E-02 | 8.624E-01 | 1.067E+00 | **1.883E-07** | 9.508E-01 | 7.087E-01 | 3.031E-01 | 7.685E+02 |
| | STD | 1.178E-02 | 3.348E+07 | 2.637E-02 | 2.080E-02 | 4.598E-01 | 1.187E-02 | 4.406E-07 | 2.848E-02 | 5.623E-02 | 7.594E-02 | 1.456E+03 |
| | Rank | 3 | 11 | 6 | 2 | 7 | 9 | 1 | 8 | 5 | 4 | 10 |
| F13_30D | Average | **3.684E-15** | 2.632E+05 | 4.446E-01 | 2.086E-01 | 2.894E+00 | 2.770E+00 | 3.014E-06 | 1.581E+00 | 2.281E+00 | 3.012E-01 | 8.914E-07 |

(*continued on next page*)

**Table 7** (*continued*)

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | STD | 1.224E-15 | 2.825E+05 | 1.813E-01 | 1.810E-01 | 2.882E-01 | 1.195E-01 | 4.439E-06 | 2.327E-01 | 3.994E-01 | 9.061E-01 | 4.568E-07 |
| | Rank | 1 | 11 | 6 | 4 | 10 | 9 | 3 | 7 | 8 | 5 | 2 |
| F13_100D | Average | **4.581E-06** | 2.507E+07 | 6.242E+00 | 1.448E+00 | 9.973E+00 | 9.927E+00 | 8.836E-06 | 8.531E+00 | 9.672E+00 | 6.641E+00 | 2.829E+01 |
| | STD | 3.384E-06 | 1.262E+07 | 3.647E-01 | 6.061E-01 | 2.645E-02 | 7.402E-02 | 1.281E-05 | 2.904E-01 | 5.570E+02 | 4.411E+00 | 2.796E+01 |
| | Rank | 1 | 11 | 5 | 3 | 9 | 8 | 2 | 6 | 7 | 4 | 10 |
| F13_500D | Average | 4.071E+01 | 6.331E+08 | 4.592E+01 | 9.980E+00 | 4.997E+01 | 5.019E+01 | **8.870E-05** | 4.825E+01 | 4.973E+01 | 4.853E+01 | 5.381E+04 |
| | STD | 1.652E+00 | 1.071E+08 | 5.916E-01 | 2.844E+00 | 4.305E-02 | 2.494E-02 | 1.307E-04 | 3.804E-01 | 1.059E-01 | 4.440E+00 | 7.706E+04 |
| | Rank | 3 | 11 | 4 | 2 | 8 | 9 | 1 | 6 | 7 | 5 | 10 |
| 30D | Mean Rank | 1.08 | 10.62 | 7.46 | 5.77 | 4.23 | 6.69 | 3.00 | 6.62 | 4.92 | 3.00 | 6.92 |
| | Final Rank | 1 | 11 | 10 | 6 | 4 | 8 | 2 | 7 | 5 | 2 | 9 |
| 100D | Mean Rank | 1.15 | 10.38 | 6.92 | 5.15 | 4.77 | 7.23 | 3.08 | 6.23 | 4.62 | 2.62 | 8.69 |
| | Final Rank | 1 | 11 | 8 | 6 | 5 | 9 | 3 | 7 | 4 | 2 | 10 |
| 500D | Mean Rank | 1.62 | 10.46 | 7.08 | 4.62 | 4.54 | 8.08 | 2.77 | 7.23 | 4.00 | 2.54 | 9.38 |
| | Final Rank | 1 | 11 | 7 | 6 | 5 | 9 | 3 | 8 | 4 | 2 | 10 |

Four types of optimization results including the best fitness values, average fitness values, standard deviation, and Friedman ranking test are presented in Table 8. Friedman ranking test shows that BSLO obtains the smallest average values, ranked the first, followed by DO, GWO, AO, GJO, SCSO, WOA, FOX, PSO, AOA, and SHO. For unimodal and multimodal functions, BSLO finds the best average optimal solutions for F6, F7, F9 and F10, ranked the first according to the number of the average optimal solutions compared to other algorithms, while for the remaining functions, the average optimal solutions calculated by BSLO are ranked the second or third, which shows the good optimization ability of BSLO. Moreover, for the hybrid and composite functions, BSLO is also ranked the first in the number of the average optimal solution (F11-F13, F17, F20-F22, F25, F27, F28, F30), and is also very competitive for most of the remaining functions, which indicates that BSLO owns better ability of local optima avoidance than other algorithms. The Wilcoxon signed-rank test results of BSLO and comparative algorithms are presented in Table 9. A significant level and the sign meaning are the same as before. It can be noted that BSLO wins against comparative algorithms for the most functions, especially for PSO, WOA, SHO, and AOA, which is consistent with the Friedman ranking test results. Therefore, BSLO shows stronger performance compared to comparative algorithms for solving CEC 2017 functions.

Fig. 7 presents the convergence curves of BSLO and comparative algorithms, which is utilized to observe the detail optimization process for CEC 2017 functions. The results show that BSLO exhibits very fast convergence speed for most functions. Compared with the other ten algorithms, BSLO obtains fastest convergence speed for F1-F7, F9, F10, F12, F13, F17, F20, F21, F23, F25, and F27-F30, and is a competitive algorithm for F8, F11, F16, F18, F19, and F22. Furthermore, BSLO also shows the competitive convergence speed at the beginning compared to the most comparative algorithms for F14, F15, F24, although BSLO loses to most comparative algorithms for these functions. In short, BSLO possesses very strong optimization performance and fast convergence speed for solving complex problems.

### 3.3. Results of BSLO for CEC 2019 test functions

CEC 2019 test functions are also applied to further assess BSLO's performance for solving complex problems, and its details can be found in Ref. [99]. BSLO is still compared with PSO, GWO, WOA, SHO, AOA, AO, GJO, SCSO, FOX and DO. The maximum iterations, number of candidate solutions, parameter values and running times are set to the same as before. The four evaluation criteria mentioned in the previous functions are also performed and shown in Table 10. It can be seen that BSLO obtains the Friedman ranking test result of 1.8, ranked the first, followed by DO with a value of 3.4, which is ranked the second. In addition, BSLO gets the best average optimal solution for F2-F7, accounting for 60%. For remaining functions, BSLO also presents competitive performance compared to ten algorithms. The Wilcoxon rank sum test results are given in Table 11. Results show that BSLO obtains better optimization results for more than 80% of CEC 2019 functions than comparative algorithms, especially for WOA, SHO and AOA.

Fig. 8 presents the convergence curve of BSLO and comparative algorithms. It is noticeable that BSLO has the fastest convergence speed compared to ten algorithms for F2-F7, and shows the competitive convergence speed for remaining functions, which is consistent with the Friedman ranking test results. Also, BSLO can converge to a very small value with few iterations for all CEC 2019 functions. Overall, BSLO can show strong optimization performance for solving various complex problems.

## 4. Engineering design problems

The practicality of BSLO for optimizing real-world problems is evaluated using five classical engineering design problems under

**Table 8**
Results of BSLO and other ten algorithms for CEC 2017 test functions (the best average optimal solution is marked in bold).

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Best | 2.973E+02 | 4.910E+05 | 6.591E+03 | 1.116E+06 | 7.149E+09 | 1.271E+09 | 9.128E+04 | 3.428E+04 | 6.913E+03 | 1.102E+02 | 1.177E+02 |
| | Average | 4.619E+03 | 3.846E+08 | 1.691E+07 | 4.834E+06 | 1.469E+10 | 6.186E+09 | 8.645E+05 | 2.335E+08 | 1.565E+08 | **2.502E+03** | 4.465E+03 |
| | STD | 4.010E+03 | 3.798E+08 | 6.441E+07 | 3.756E+06 | 5.183E+09 | 2.930E+09 | 5.412E+05 | 2.589E+08 | 3.108E+08 | 3.513E+03 | 3.022E+03 |
| F3 | Best | 3.000E+02 | 2.582E+03 | 3.048E+02 | 5.642E+02 | 1.781E+04 | 3.596E+03 | 3.133E+02 | 4.471E+02 | 3.273E+02 | 3.000E+02 | 3.000E+02 |
| | Average | 3.000E+02 | 9.126E+03 | 1.935E+03 | 3.224E+03 | 5.486E+04 | 9.557E+03 | 7.727E+02 | 4.099E+03 | 2.259E+03 | **3.000E+02** | 3.000E+02 |
| | STD | 6.666E-03 | 6.479E+03 | 1.981E+03 | 3.548E+03 | 3.639E+04 | 3.144E+03 | 3.695E+02 | 3.249E+03 | 2.073E+03 | 9.546E-05 | 1.051E-02 |
| F4 | Best | 4.001E+02 | 4.084E+02 | 4.000E+02 | 4.005E+02 | 7.485E+02 | 5.870E+02 | 4.003E+02 | 4.068E+02 | 4.009E+02 | 4.001E+02 | 4.000E+02 |
| | Average | 4.069E+02 | 4.937E+02 | 4.180E+02 | 4.289E+02 | 1.945E+03 | 9.200E+02 | 4.194E+02 | 4.387E+02 | 4.333E+02 | **4.040E+02** | 4.049E+02 |
| | STD | 1.142E+01 | 6.674E+01 | 1.887E+01 | 2.795E+01 | 6.532E+02 | 2.748E+02 | 2.286E+01 | 2.951E+01 | 2.991E+01 | 6.733E+00 | 2.762E+00 |
| F5 | Best | 5.090E+02 | 5.199E+02 | 5.041E+02 | 5.185E+02 | 6.059E+02 | 5.246E+02 | 5.112E+02 | 5.131E+02 | 5.107E+02 | 5.249E+02 | 5.080E+02 |
| | Average | 5.164E+02 | 5.400E+02 | **5.161E+02** | 5.527E+02 | 6.399E+02 | 5.477E+02 | 5.277E+02 | 5.300E+02 | 5.354E+02 | 5.774E+02 | 5.290E+02 |
| | STD | 6.274E+00 | 1.384E+01 | 7.126E+00 | 2.011E+01 | 1.879E+01 | 1.757E+01 | 9.423E+00 | 1.104E+01 | 1.405E+01 | 2.487E+01 | 1.095E+01 |
| F6 | Best | 6.000E+02 | 6.101E+02 | 6.001E+02 | 6.107E+02 | 6.523E+02 | 6.228E+02 | 6.059E+02 | 6.028E+02 | 6.028E+02 | 6.383E+02 | 6.002E+02 |
| | Average | **6.000E+02** | 6.263E+02 | 6.009E+02 | 6.366E+02 | 6.838E+02 | 6.371E+02 | 6.141E+02 | 6.085E+02 | 6.149E+02 | 6.540E+02 | 6.046E+02 |
| | STD | 3.177E-04 | 9.881E+00 | 1.387E+00 | 1.181E+01 | 1.478E+01 | 6.803E+00 | 5.269E+00 | 5.963E+00 | 9.170E+00 | 7.275E+00 | 4.507E+00 |
| F7 | Best | 7.154E+02 | 7.217E+02 | 7.162E+02 | 7.421E+02 | 8.385E+02 | 7.758E+02 | 7.262E+02 | 7.240E+02 | 7.322E+02 | 8.056E+02 | 7.213E+02 |
| | Average | **7.269E+02** | 7.518E+02 | 7.297E+02 | 7.819E+02 | 8.642E+02 | 7.980E+02 | 7.506E+02 | 7.486E+02 | 7.655E+02 | 8.146E+02 | 7.496E+02 |
| | STD | 6.115E+00 | 1.750E+01 | 9.924E+00 | 2.611E+01 | 1.746E+01 | 1.252E+01 | 1.587E+01 | 1.526E+01 | 1.644E+01 | 5.249E+00 | 1.299E+01 |
| F8 | Best | 8.050E+02 | 8.138E+02 | 8.048E+02 | 8.162E+02 | 8.727E+02 | 8.170E+02 | 8.052E+02 | 8.142E+02 | 8.060E+02 | 8.179E+02 | 8.080E+02 |
| | Average | 8.173E+02 | 8.276E+02 | **8.147E+02** | 8.395E+02 | 9.018E+02 | 8.312E+02 | 8.240E+02 | 8.268E+02 | 8.279E+02 | 8.452E+02 | 8.239E+02 |
| | STD | 8.281E+00 | 9.757E+00 | 8.102E+00 | 1.701E+01 | 1.652E+01 | 6.442E+00 | 7.793E+00 | 8.142E+00 | 9.064E+00 | 1.826E+01 | 8.036E+00 |
| F9 | Best | 9.000E+02 | 9.289E+02 | 9.000E+02 | 9.574E+02 | 1.974E+03 | 1.020E+03 | 9.226E+02 | 9.080E+02 | 9.082E+02 | 1.266E+03 | 9.000E+02 |
| | Average | **9.005E+02** | 1.152E+03 | 9.104E+02 | 1.435E+03 | 2.669E+03 | 1.350E+03 | 1.024E+03 | 9.768E+02 | 1.061E+03 | 1.765E+03 | 9.164E+02 |
| | STD | 1.847E+00 | 1.879E+02 | 1.707E+01 | 4.839E+02 | 4.967E+02 | 1.769E+02 | 7.401E+01 | 5.414E+01 | 1.316E+02 | 1.678E+02 | 3.496E+01 |
| F10 | Best | 1.135E+03 | 1.423E+03 | 1.130E+03 | 1.564E+03 | 2.943E+03 | 1.460E+03 | 1.352E+03 | 1.244E+03 | 1.141E+03 | 1.853E+03 | 1.256E+03 |
| | Average | **1.546E+03** | 2.108E+03 | 1.574E+03 | 2.181E+03 | 3.457E+03 | 2.086E+03 | 1.886E+03 | 1.979E+03 | 1.932E+03 | 2.415E+03 | 1.807E+03 |
| | STD | 2.077E+02 | 3.588E+02 | 2.247E+02 | 2.852E+02 | 2.627E+02 | 3.123E+02 | 2.926E+02 | 4.973E+02 | 3.429E+02 | 3.911E+02 | 3.244E+02 |
| F11 | Best | 1.104E+03 | 1.116E+03 | 1.109E+03 | 1.122E+03 | 2.496E+03 | 1.172E+03 | 1.123E+03 | 1.121E+03 | 1.112E+03 | 1.118E+03 | 1.105E+03 |
| | Average | **1.118E+03** | 1.310E+03 | 1.144E+03 | 1.229E+03 | 1.254E+04 | 1.949E+03 | 1.194E+03 | 1.185E+03 | 1.157E+03 | 1.217E+03 | 1.123E+03 |
| | STD | 2.589E+01 | 3.416E+02 | 5.073E+01 | 9.576E+01 | 1.068E+04 | 1.120E+03 | 4.740E+01 | 7.315E+01 | 4.712E+01 | 6.536E+01 | 9.363E+00 |
| F12 | Best | 2.141E+03 | 9.209E+03 | 9.855E+03 | 8.796E+04 | 1.391E+08 | 2.480E+04 | 9.716E+03 | 2.168E+04 | 4.337E+04 | 1.001E+04 | 1.038E+04 |
| | Average | **2.541E+04** | 3.124E+06 | 5.661E+05 | 5.663E+06 | 1.219E+09 | 4.065E+07 | 2.579E+06 | 1.025E+06 | 1.000E+06 | 2.603E+05 | 3.807E+05 |
| | STD | 2.248E+04 | 3.643E+06 | 7.300E+05 | 5.330E+06 | 8.138E+08 | 9.129E+07 | 2.809E+06 | 1.299E+06 | 9.245E+05 | 2.761E+05 | 5.475E+05 |
| F13 | Best | 1.313E+03 | 2.326E+03 | 2.029E+03 | 2.051E+03 | 9.902E+06 | 3.625E+03 | 3.134E+03 | 3.614E+03 | 2.395E+03 | 1.560E+03 | 2.130E+03 |
| | Average | **7.738E+03** | 1.382E+04 | 1.140E+04 | 1.509E+04 | 1.170E+08 | 9.329E+03 | 1.346E+04 | 1.090E+04 | 1.457E+04 | 1.584E+04 | 8.937E+03 |
| | STD | 8.573E+03 | 7.524E+03 | 7.675E+03 | 1.203E+04 | 1.650E+08 | 6.143E+03 | 9.116E+03 | 5.398E+03 | 8.988E+03 | 1.461E+04 | 7.568E+03 |
| F14 | Best | 1.432E+03 | 1.488E+03 | 1.452E+03 | 1.480E+03 | 4.470E+03 | 1.508E+03 | 1.510E+03 | 1.461E+03 | 1.456E+03 | 1.582E+03 | 1.445E+03 |
| | Average | 3.571E+03 | 6.552E+03 | 3.126E+03 | 2.284E+03 | 3.896E+06 | 7.945E+03 | 1.841E+03 | 2.546E+03 | 2.695E+03 | 5.870E+03 | **1.620E+03** |
| | STD | 2.337E+03 | 4.796E+03 | 1.770E+03 | 1.212E+03 | 1.144E+07 | 6.417E+03 | 3.456E+02 | 1.693E+03 | 1.736E+03 | 5.501E+03 | 2.855E+02 |
| F15 | Best | 1.529E+03 | 2.008E+03 | 1.553E+03 | 1.656E+03 | 7.465E+03 | 5.897E+03 | 2.046E+03 | 1.769E+03 | 1.572E+03 | 1.877E+03 | 1.563E+03 |
| | Average | 5.138E+03 | 2.033E+04 | 4.576E+03 | 7.061E+03 | 2.110E+06 | 1.736E+04 | 4.875E+03 | 3.910E+03 | 3.697E+03 | 2.259E+04 | **2.203E+03** |
| | STD | 6.410E+03 | 2.323E+04 | 2.663E+03 | 5.088E+03 | 3.659E+06 | 5.417E+03 | 2.223E+03 | 1.938E+03 | 1.674E+03 | 1.916E+04 | 1.392E+03 |
| F16 | Best | 1.612E+03 | 1.634E+03 | 1.612E+03 | 1.637E+03 | 2.197E+03 | 1.640E+03 | 1.623E+03 | 1.634E+03 | 1.613E+03 | 1.857E+03 | 1.604E+03 |
| | Average | 1.754E+03 | 1.919E+03 | **1.703E+03** | 1.898E+03 | 2.538E+03 | 2.023E+03 | 1.790E+03 | 1.771E+03 | 1.776E+03 | 2.240E+03 | 1.775E+03 |
| | STD | 1.028E+02 | 1.487E+02 | 8.661E+01 | 1.395E+02 | 2.240E+02 | 1.718E+02 | 1.230E+02 | 1.290E+02 | 1.306E+02 | 1.939E+02 | 1.246E+02 |
| F17 | Best | 1.701E+03 | 1.732E+03 | 1.722E+03 | 1.735E+03 | 1.847E+03 | 1.761E+03 | 1.737E+03 | 1.736E+03 | 1.731E+03 | 1.750E+03 | 1.720E+03 |
| | Average | **1.746E+03** | 1.815E+03 | 1.762E+03 | 1.802E+03 | 2.216E+03 | 1.867E+03 | 1.766E+03 | 1.763E+03 | 1.761E+03 | 2.018E+03 | 1.756E+03 |
| | STD | 3.798E+01 | 8.795E+01 | 3.592E+01 | 5.307E+01 | 2.091E+02 | 8.449E+01 | 1.872E+01 | 1.501E+01 | 2.010E+01 | 2.074E+02 | 2.315E+01 |
| F18 | Best | 3.114E+03 | 2.003E+03 | 2.459E+03 | 2.448E+03 | 1.341E+07 | 2.401E+03 | 6.537E+03 | 4.136E+03 | 4.199E+03 | 3.290E+03 | 2.900E+03 |
| | Average | 1.761E+04 | **1.220E+04** | 2.613E+04 | 1.588E+04 | 3.348E+08 | 1.670E+04 | 2.283E+04 | 3.613E+04 | 2.014E+04 | 1.610E+04 | 2.095E+04 |
| | STD | 1.266E+04 | 9.194E+03 | 1.413E+04 | 1.160E+04 | 3.007E+08 | 1.088E+04 | 1.145E+04 | 9.831E+03 | 1.277E+04 | 1.092E+04 | 1.226E+04 |
| F19 | Best | 1.908E+03 | 2.052E+03 | 1.922E+03 | 2.825E+03 | 3.137E+03 | 2.230E+03 | 2.517E+03 | 1.937E+03 | 1.924E+03 | 2.009E+03 | 1.907E+03 |
| | Average | 7.204E+03 | 1.457E+04 | 8.908E+03 | 4.005E+04 | 1.740E+07 | 3.664E+04 | 8.446E+03 | 8.783E+03 | 5.149E+03 | 7.935E+03 | **3.540E+03** |
| | STD | 7.376E+03 | 2.296E+04 | 6.182E+03 | 4.580E+04 | 2.522E+07 | 2.953E+04 | 9.142E+03 | 5.643E+03 | 4.965E+03 | 6.898E+03 | 3.284E+03 |

(*continued on next page*)

**Table 8** (*continued*)

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F20 | Best | 2.001E+03 | 2.024E+03 | 2.022E+03 | 2.053E+03 | 2.203E+03 | 2.054E+03 | 2.057E+03 | 2.039E+03 | 2.020E+03 | 2.066E+03 | 2.022E+03 |
| | Average | **2.015E+03** | 2.145E+03 | 2.063E+03 | 2.181E+03 | 2.450E+03 | 2.152E+03 | 2.121E+03 | 2.104E+03 | 2.020E+03 | 2.258E+03 | 2.063E+03 |
| | STD | 9.969E+00 | 5.864E+01 | 4.094E+01 | 8.850E+01 | 1.241E+02 | 7.991E+01 | 5.957E+01 | 5.216E+01 | 7.742E+01 | 1.322E+02 | 3.884E+01 |
| F21 | Best | 2.200E+03 | 2.212E+03 | 2.201E+03 | 2.213E+03 | 2.313E+03 | 2.235E+03 | 2.207E+03 | 2.205E+03 | 2.200E+03 | 2.200E+03 | 2.200E+03 |
| | Average | **2.241E+03** | 2.315E+03 | 2.304E+03 | 2.322E+03 | 2.411E+03 | 2.315E+03 | 2.286E+03 | 2.314E+03 | 2.294E+03 | 2.337E+03 | 2.317E+03 |
| | STD | 6.027E+01 | 4.877E+01 | 3.512E+01 | 4.970E+01 | 4.295E+01 | 3.944E+01 | 5.568E+01 | 4.082E+01 | 5.943E+01 | 7.276E+01 | 4.812E+01 |
| F22 | Best | 2.238E+03 | 2.226E+03 | 2.202E+03 | 2.237E+03 | 2.954E+03 | 2.386E+03 | 2.248E+03 | 2.227E+03 | 2.301E+03 | 2.304E+03 | 2.300E+03 |
| | Average | **2.297E+03** | 2.428E+03 | 2.304E+03 | 2.371E+03 | 3.829E+03 | 2.876E+03 | 2.305E+03 | 2.365E+03 | 2.319E+03 | 2.484E+03 | 2.304E+03 |
| | STD | 1.973E+01 | 3.554E+02 | 2.564E+01 | 3.090E+02 | 5.614E+02 | 2.658E+02 | 1.704E+01 | 6.562E+01 | 2.321E+01 | 4.355E+02 | 3.122E+00 |
| F23 | Best | 2.611E+03 | 2.642E+03 | 2.603E+03 | 2.618E+03 | 2.698E+03 | 2.651E+03 | 2.616E+03 | 2.616E+03 | 2.617E+03 | 2.614E+03 | 2.615E+03 |
| | Average | 2.624E+03 | 2.700E+03 | **2.620E+03** | 2.650E+03 | 2.776E+03 | 2.730E+03 | 2.636E+03 | 2.634E+03 | 2.637E+03 | 2.751E+03 | 2.640E+03 |
| | STD | 8.570E+00 | 3.807E+01 | 8.973E+00 | 2.412E+01 | 5.501E+01 | 4.887E+01 | 1.253E+01 | 1.050E+01 | 1.527E+01 | 6.391E+01 | 1.434E+01 |
| F24 | Best | 2.500E+03 | 2.541E+03 | 2.519E+03 | 2.579E+03 | 2.742E+03 | 2.677E+03 | 2.501E+03 | 2.511E+03 | 2.501E+03 | 2.500E+03 | 2.500E+03 |
| | Average | 2.753E+03 | 2.764E+03 | 2.738E+03 | 2.777E+03 | 2.933E+03 | 2.837E+03 | 2.733E+03 | **2.511E+03** | 2.699E+03 | 2.880E+03 | 2.746E+03 |
| | STD | 7.052E+01 | 1.001E+02 | 4.306E+01 | 4.457E+01 | 7.566E+01 | 7.437E+01 | 9.343E+01 | 4.898E+01 | 1.207E+02 | 1.394E+02 | 9.164E+01 |
| F25 | Best | 2.600E+03 | 2.899E+03 | 2.898E+03 | 2.647E+03 | 3.197E+03 | 3.019E+03 | 2.898E+03 | 2.680E+03 | 2.901E+03 | 2.700E+03 | 2.898E+03 |
| | Average | **2.916E+03** | 2.964E+03 | 2.933E+03 | 2.942E+03 | 3.903E+03 | 3.261E+03 | 2.929E+03 | 2.930E+03 | 2.945E+03 | 2.931E+03 | 2.922E+03 |
| | STD | 7.848E+01 | 3.607E+01 | 1.938E+01 | 6.330E+01 | 4.283E+02 | 1.706E+02 | 2.378E+01 | 5.312E+01 | 2.703E+01 | 6.402E+01 | 2.513E+01 |
| F26 | Best | 2.800E+03 | 2.836E+03 | 2.817E+03 | 2.694E+03 | 3.976E+03 | 3.336E+03 | 2.608E+03 | 2.977E+03 | 2.604E+03 | 2.900E+03 | 2.800E+03 |
| | Average | 3.047E+03 | 3.503E+03 | 3.003E+03 | 3.461E+03 | 4.647E+03 | 3.987E+03 | **2.977E+03** | 3.106E+03 | 3.087E+03 | 4.024E+03 | 3.101E+03 |
| | STD | 2.588E+02 | 4.563E+02 | 2.304E+02 | 6.366E+02 | 3.515E+02 | 2.946E+02 | 1.854E+02 | 1.952E+02 | 2.712E+02 | 7.073E+02 | 3.560E+02 |
| F27 | Best | 3.090E+03 | 3.123E+03 | 3.090E+03 | 3.099E+03 | 3.168E+03 | 3.160E+03 | 3.092E+03 | 3.090E+03 | 3.090E+03 | 3.131E+03 | 3.091E+03 |
| | Average | **3.096E+03** | 3.199E+03 | 3.097E+03 | 3.134E+03 | 3.321E+03 | 3.248E+03 | 3.102E+03 | 3.104E+03 | 3.103E+03 | 3.225E+03 | 3.106E+03 |
| | STD | 4.441E+00 | 4.143E+01 | 4.680E+00 | 3.649E+01 | 9.363E+01 | 7.352E+01 | 4.656E+00 | 1.779E+01 | 1.803E+01 | 5.529E+01 | 2.018E+01 |
| F28 | Best | 2.800E+03 | 3.192E+03 | 3.169E+03 | 3.214E+03 | 3.672E+03 | 3.383E+03 | 3.182E+03 | 3.173E+03 | 3.100E+03 | 3.100E+03 | 3.100E+03 |
| | Average | **3.276E+03** | 3.535E+03 | 3.370E+03 | 3.429E+03 | 3.927E+03 | 3.753E+03 | 3.386E+03 | 3.378E+03 | 3.346E+03 | 3.297E+03 | 3.293E+03 |
| | STD | 1.711E+02 | 1.414E+02 | 1.007E+02 | 1.809E+02 | 1.377E+02 | 1.619E+02 | 9.915E+01 | 1.239E+02 | 1.190E+02 | 1.329E+02 | 1.251E+02 |
| F29 | Best | 3.145E+03 | 3.172E+03 | 3.133E+03 | 3.189E+03 | 3.424E+03 | 3.223E+03 | 3.185E+03 | 3.154E+03 | 3.160E+03 | 3.191E+03 | 3.145E+03 |
| | Average | 3.206E+03 | 3.315E+03 | 3.203E+03 | 3.351E+03 | 3.796E+03 | 3.367E+03 | 3.233E+03 | **3.198E+03** | 3.237E+03 | 3.535E+03 | 3.260E+03 |
| | STD | 5.587E+01 | 9.412E+01 | 5.380E+01 | 9.015E+01 | 2.114E+02 | 9.285E+01 | 4.320E+01 | 2.912E+01 | 6.573E+01 | 1.767E+02 | 6.908E+01 |
| F30 | Best | 5.293E+03 | 5.937E+04 | 8.136E+03 | 1.162E+04 | 3.047E+06 | 1.141E+04 | 4.239E+03 | 7.431E+03 | 8.151E+03 | 1.098E+04 | 6.822E+03 |
| | Average | **1.887E+05** | 5.909E+06 | 7.252E+05 | 1.034E+06 | 6.780E+07 | 1.901E+07 | 5.447E+05 | 8.387E+05 | 1.136E+06 | 8.890E+05 | 2.912E+05 |
| | STD | 3.333E+05 | 7.880E+06 | 8.905E+05 | 9.801E+05 | 5.920E+07 | 1.553E+07 | 6.603E+05 | 1.179E+06 | 1.255E+06 | 1.706E+06 | 5.073E+05 |
| Mean | Rank | **2.21** | **7.62** | 3.59 | 7.45 | 11 | 8.69 | 4.48 | 4.93 | 5.1 | 7.55 | 3.38 |
| Final | Ranking | 1 | 9 | 3 | 7 | 11 | 10 | 4 | 5 | 6 | 8 | 2 |

**Table 9**
The Wilcoxon rank sum text between BSLO and other ten algorithms for CEC 2017 test functions.

| F | PSO p (Sig.) | GWO p (Sig.) | WOA p (Sig.) | SHO p (Sig.) | AOA p (Sig.) | AO p (Sig.) | GJO p (Sig.) | SCSO p (Sig.) | FOX p (Sig.) | DO p (Sig.) |
|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 3.02E-11 (+) | 2.61E-10 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 6.01E-08 (+) | 1.33E-04 (-) | 2.51E-02 (-) |
| F3 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 7.04E-07 (-) | 7.62E-03 (+) |
| F4 | 7.38E-10 (+) | 6.53E-08 (+) | 3.08E-08 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 1.17E-04 (+) | 8.48E-09 (+) | 1.49E-06 (+) | 3.85E-03 (-) | 6.74E-01 (~) |
| F5 | 1.87E-07 (+) | 2.84E-01 (~) | 4.57E-09 (+) | 3.02E-11 (+) | 9.92E-11 (+) | 1.89E-04 (+) | 3.16E-05 (+) | 2.00E-06 (+) | 3.02E-11 (+) | 5.08E-03 (+) |
| F6 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F7 | 5.61E-05 (+) | 2.07E-02 (+) | 4.20E-10 (+) | 3.02E-11 (+) | 3.34E-11 (+) | 2.23E-09 (+) | 8.48E-09 (+) | 3.52E-07 (+) | 3.02E-11 (+) | 5.09E-08 (+) |
| F8 | 2.61E-02 (+) | 2.06E-01 (~) | 6.53E-08 (+) | 3.02E-11 (+) | 2.39E-08 (+) | 9.26E-09 (+) | 1.75E-05 (+) | 1.49E-06 (+) | 3.34E-11 (+) | 1.73E-07 (+) |
| F9 | 6.70E-11 (+) | 5.19E-07 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 5.49E-11 (+) | 2.61E-10 (+) | 7.39E-11 (+) | 3.02E-11 (+) | 7.09E-08 (+) |
| F10 | 9.06E-08 (+) | 8.77E-02 (+) | 2.23E-09 (+) | 3.02E-11 (+) | 1.85E-08 (+) | 4.46E-04 (+) | 6.38E-03 (+) | 1.86E-06 (+) | 1.21E-10 (+) | 2.60E-05 (+) |
| F11 | 9.26E-09 (+) | 4.12E-06 (+) | 4.20E-10 (+) | 3.02E-11 (+) | 6.70E-11 (+) | 1.69E-09 (+) | 3.82E-10 (+) | 2.03E-09 (+) | 7.39E-11 (+) | 3.85E-03 (+) |
| F12 | 2.87E-10 (+) | 3.83E-05 (+) | 8.10E-10 (+) | 3.02E-11 (+) | 4.57E-09 (+) | 5.57E-10 (+) | 3.03E-03 (+) | 1.86E-09 (+) | 8.35E-08 (+) | 4.42E-06 (+) |
| F13 | 2.61E-10 (+) | 4.12E-01 (+) | 2.32E-02 (+) | 3.02E-11 (+) | 8.42E-01 (~) | 1.08E-02 (+) | 4.83E-01 (~) | 4.51E-02 (+) | 1.08E-02 (+) | 6.84E-01 (~) |
| F14 | 2.28E-01 (~) | 3.39E-02 (-) | 1.06E-03 (-) | 2.39E-08 (+) | 2.24E-02 (+) | 9.52E-04 (-) | 6.91E-04 (-) | 1.50E-02 (-) | 7.51E-01 (~) | 3.99E-04 (-) |
| F15 | 2.75E-03 (+) | 7.51E-01 (~) | 3.78E-02 (+) | 4.50E-11 (+) | 1.86E-09 (+) | 1.58E-01 (~) | 9.82E-01 (~) | 2.97E-01 (~) | 5.46E-06 (+) | 2.38E-03 (-) |
| F16 | 6.74E-06 (+) | 6.35E-02 (~) | 4.46E-04 (+) | 3.02E-11 (+) | 2.88E-06 (+) | 5.75E-02 (~) | 7.30E-04 (+) | 1.38E-02 (+) | 2.78E-07 (+) | 3.11E-01 (~) |
| F17 | 4.74E-06 (+) | 9.52E-04 (+) | 8.88E-06 (+) | 4.50E-11 (+) | 3.96E-08 (+) | 4.64E-05 (+) | 3.01E-04 (+) | 3.59E-05 (+) | 7.77E-09 (+) | 9.47E-03 (+) |
| F18 | 7.74E-06 (-) | 3.64E-02 (+) | 7.01E-02 (~) | 3.02E-11 (+) | 2.92E-02 (-) | 3.85E-03 (+) | 1.04E-04 (+) | 8.53E-01 (~) | 4.68E-02 (-) | 2.51E-02 (+) |
| F19 | 3.87E-01 (~) | 6.10E-03 (+) | 9.88E-03 (+) | 3.02E-11 (+) | 8.20E-07 (+) | 5.69E-01 (~) | 4.43E-03 (+) | 8.77E-02 (~) | 6.35E-02 (~) | 7.96E-03 (-) |
| F20 | 3.02E-11 (+) | 2.67E-09 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.69E-11 (+) | 3.02E-11 (+) | 1.43E-08 (+) |
| F21 | 5.56E-04 (+) | 1.00E-03 (+) | 1.19E-06 (+) | 3.47E-10 (+) | 2.05E-03 (+) | 4.71E-04 (+) | 9.07E-03 (+) | 4.21E-02 (+) | 4.31E-08 (+) | 2.92E-02 (+) |
| F22 | 2.19E-08 (+) | 2.62E-03 (~) | 5.97E-09 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 7.70E-08 (+) | 1.96E-10 (+) | 3.18E-03 (+) | 7.04E-07 (+) | 3.03E-03 (+) |
| F23 | 7.38E-10 (+) | 3.63E-01 (~) | 6.05E-07 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.81E-07 (+) | 5.26E-04 (+) | 9.79E-05 (+) | 3.34E-11 (+) | 6.28E-06 (+) |
| F24 | 1.96E-10 (+) | 2.03E-07 (-) | 8.31E-03 (+) | 4.98E-11 (+) | 1.09E-05 (+) | 1.37E-01 (~) | 3.48E-01 (~) | 9.82E-01 (~) | 5.49E-11 (+) | 1.76E-02 (-) |
| F25 | 2.51E-02 (+) | 3.39E-02 (+) | 3.18E-03 (+) | 3.02E-11 (+) | 2.61E-10 (+) | 9.23E-01 (~) | 4.92E-01 (~) | 2.17E-01 (~) | 3.92E-02 (+) | 2.05E-03 (+) |
| F26 | 7.30E-04 (+) | 1.37E-01 (~) | 3.32E-06 (+) | 4.98E-11 (+) | 5.46E-09 (+) | 5.89E-01 (~) | 2.05E-03 (+) | 2.50E-03 (+) | 7.77E-09 (+) | 4.38E-01 (~) |
| F27 | 1.96E-10 (+) | 4.68E-02 (+) | 8.35E-08 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 7.96E-03 (+) | 5.32E-03 (+) | 2.32E-02 (+) | 4.08E-11 (+) | 2.84E-04 (+) |
| F28 | 1.84E-02 (+) | 2.13E-04 (+) | 5.25E-05 (+) | 3.02E-11 (+) | 1.78E-10 (+) | 3.26E-07 (+) | 1.63E-02 (+) | 6.36E-05 (+) | 1.38E-02 (+) | 1.81E-01 (~) |
| F29 | 4.44E-07 (+) | 3.63E-01 (~) | 4.62E-10 (+) | 3.02E-11 (+) | 2.37E-10 (+) | 1.89E-04 (+) | 6.00E-01 (~) | 2.42E-02 (+) | 3.82E-09 (+) | 9.47E-03 (+) |
| F30 | 1.39E-06 (+) | 4.84E-02 (+) | 1.44E-03 (+) | 3.02E-11 (+) | 1.31E-08 (+) | 4.21E-02 (+) | 4.36E-02 (+) | 4.86E-03 (+) | 2.39E-04 (+) | 8.24E-02 (~) |
| (W\|L\|T) | (26\|1\|2) | (20\|2\|7) | (27\|1\|1) | (29\|0\|0) | (27\|1\|1) | (22\|1\|6) | (23\|1\|5) | (23\|1\|5) | (23\|4\|2) | (18\|5\|6) |

multiple inequality constraints. These problems include tension/compression spring design problem, pressure vessel design problem, welded beam design problem, speed reducer design problem, and three-bar truss design problem. A simple death penalty is performed to transform a problem with constraints into a problem without constraints. When the candidate solution does not satisfy the constraints, a significant large value is given and then discarded in the next iteration. BSLO is compared with the algorithms mentioned before, and the maximum iteration and number of candidate solutions are fixed as 500 and 30, respectively. Each algorithm is run 30 times for each engineering problem.

### 4.1. Tension/compression spring design problem

Minimizing weight of tension/compression spring is the purpose of this problem [69]. Its structure is presented in Fig. 9. Three variables involving wire diameter ($d$), mean coil diameter ($D$), and the number of active coils ($P$) are designed to satisfy its four constraints. The involved formulation is given as follows:

Three variables: $x_1 = d$, $x_2 = D$, $x_3 = P$

Minimize:

$$f(x) = (x_3 + 2)x_2 x_1^2$$

Subject to:

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

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0,$$

$$g_3(x) = 1 - \frac{140.54 x_1}{x_2^2 x_3} \le 0,$$

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

Variables range:

$$0.05 \le x_1 \le 2,\ 0.25 \le x_2 \le 1.3,\ 2 \le x_3 \le 15$$

Table 12 shows the optimization results of BSLO and ten comparative algorithms for this design problem. When $x_1$, $x_2$ and $x_3$ are equal to 0.051669, 0.356226 and 11.31788, the optimum weight searched by BSLO is 0.0126652. It can be seen that BSLO wins all comparative algorithms for this problem.

### 4.2. Pressure vessel design problem

Pressure vessel design problem with four constraints is to search for the minimum cost using suitable variables [69]. Fig. 10 presents its structure, which involves four variables: the shell thickness ($T_s$), the head thickness ($T_h$), the inner radius ($R$) and the length without the head ($L$). The involved formulation is given as follows:

Four variables:

$$x_1 = T_s,\ x_2 = T_h,\ x_3 = R,\ x_4 = L$$

Minimize:

$$f(x) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3$$

Subject to:

$$g_1(x) = -x_1 + 0.0193 x_3 \le 0,$$

$$g_2(x) = -x_2 + 0.00954 x_3 \le 0,$$

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

$$g_4(x) = x_4 - 240 \le 0,$$

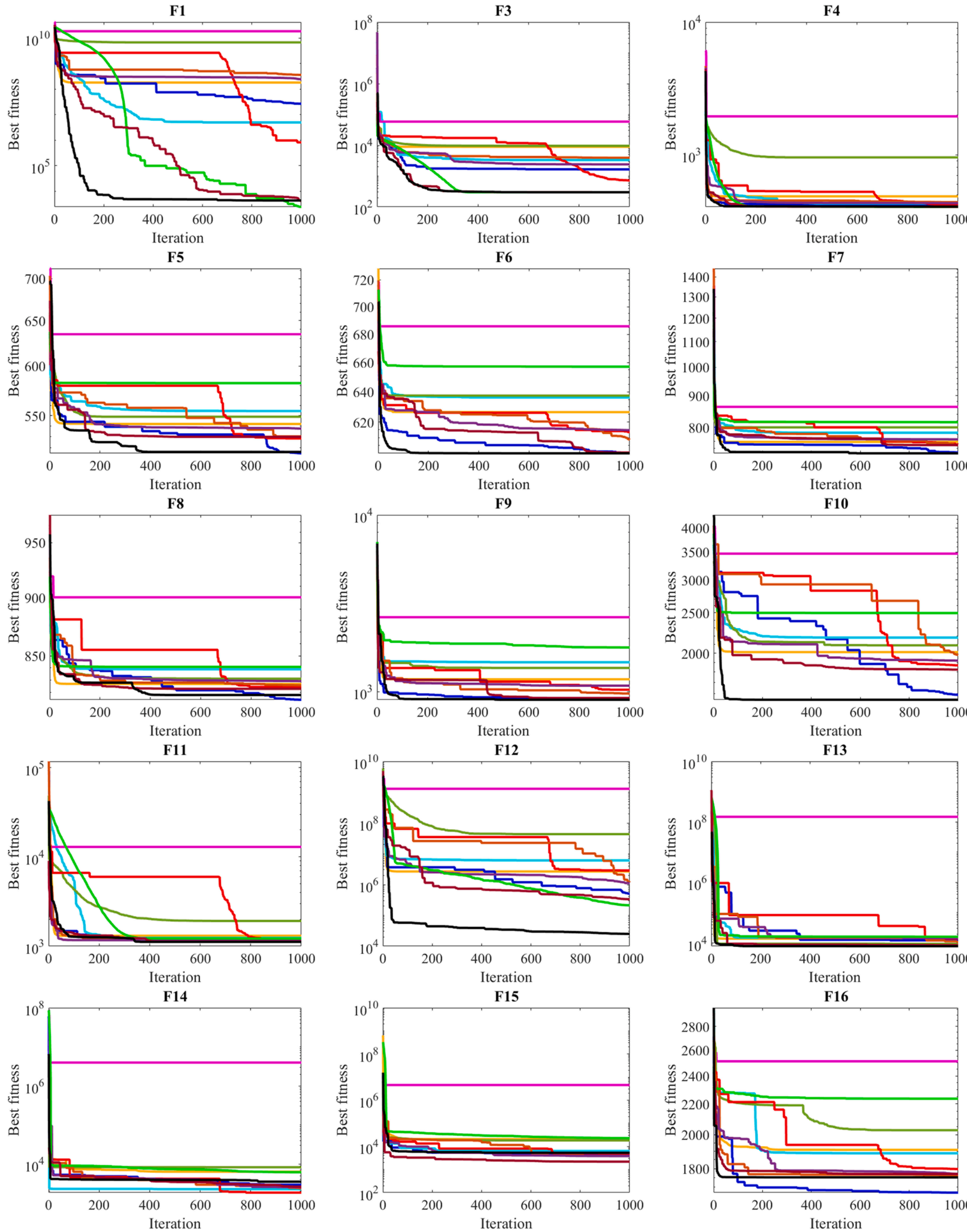


**Fig. 7.** Convergence curves of BSLO and comparative algorithms for CEC 2017 functions.

Variables range:

$$0 \le x_1 \le 99,\ 0 \le x_2 \le 99,\ 10 \le x_3 \le 200,\ 10 \le x_4 \le 200$$

The optimization results of BSLO and ten comparative algorithms are shown in Table 13. The optimum cost obtained by BSLO is 5885.3328, and corresponding values of $x_1$, $x_2$, $x_3$ and $x_4$ are 0.778169, 0.3846492, 40.31962 and 200, respectively. Also, BSLO outperforms all comparative algorithms, especially for PSO, WOA, SHO, AOA and AO.

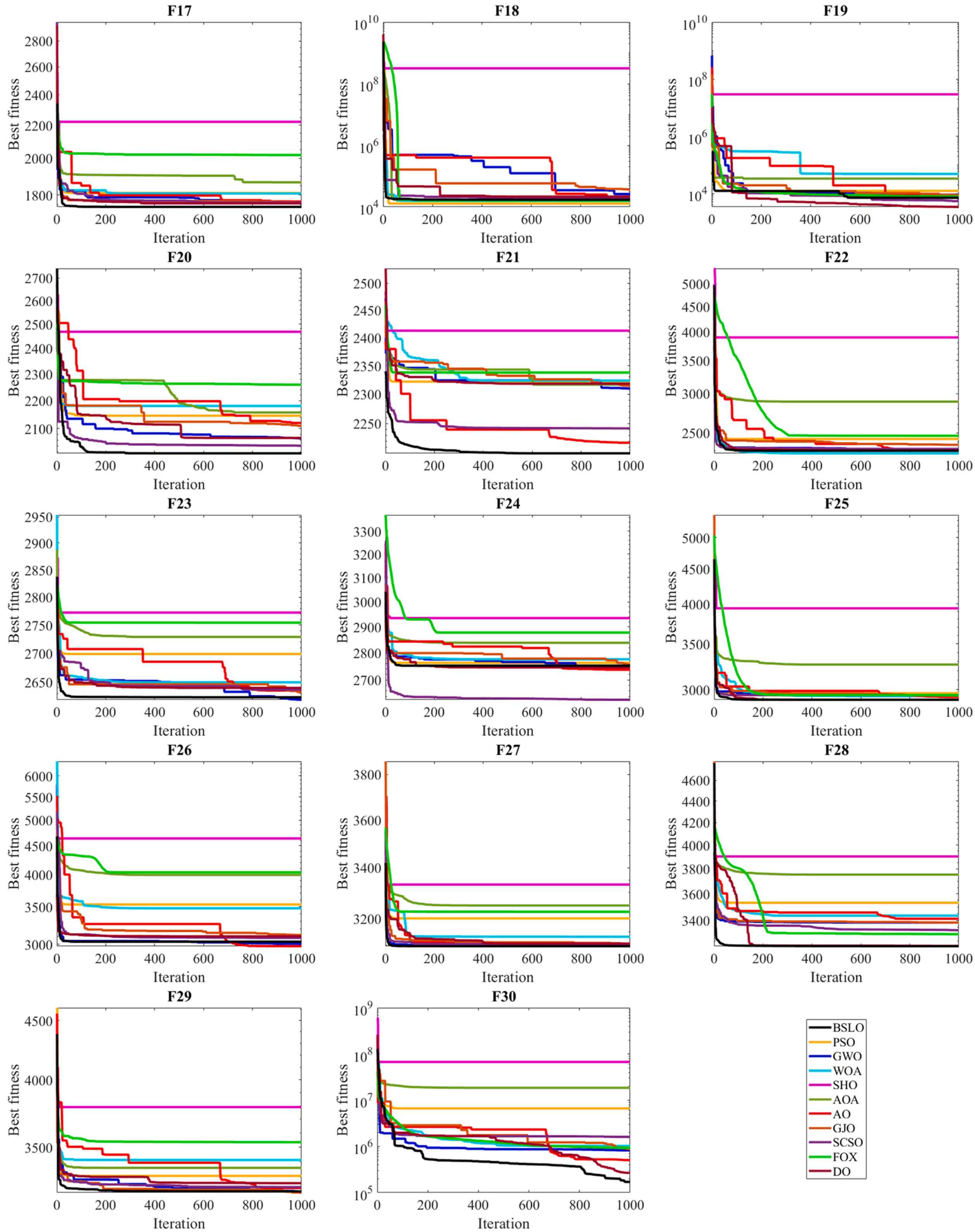


**Fig. 7.** (*continued*).

### 4.3. Welded beam design problem

Minimizing cost is the objective of welded beam design problem [69]. This problem shown in Fig. 11 includes seven constraints such as the deflection ($\delta$), shear stress ($\tau$), the bar's buckling load ($P_c$) and bending stress in beam ($\theta$), and the four design variables including the thickness of the weld ($h$), the length of the weld ($l$), the height of the bar ($t$) and the thickness of the bar ($b$). These variables are continuously optimized by BSLO and other algorithms to satisfy these seven constraints, and then the final minimum cost can be obtained. The involved formulation is given as follows:

Four variables:

**Table 10**
Results of BSLO and other ten algorithms for CEC 2019 functions (the best average optimal solution is marked in bold).

| Fun | Measure | BSLO | PSO | GWO | WOA | SHO | AOA | AO | GJO | SCSO | FOX | DO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Best | 3.88E+04 | 1.43E+08 | 3.78E+05 | 3.22E+06 | 8.51E+04 | 1.53E+05 | 4.38E+04 | 3.80E+04 | 3.90E+04 | 0.00E+00 | 1.04E+06 |
| | Average | 4.06E+04 | 2.78E+10 | 1.01E+08 | 1.67E+10 | 3.60E+05 | 5.41E+06 | 5.06E+04 | 1.21E+05 | 4.42E+04 | **1.31E+04** | 1.29E+08 |
| | STD | 1.29E+03 | 5.84E+10 | 2.17E+08 | 2.49E+10 | 3.40E+05 | 2.00E+07 | 4.69E+03 | 3.69E+05 | 3.52E+03 | 2.04E+04 | 1.68E+08 |
| F2 | Best | 17.3429 | 17.3429 | 17.3431 | 17.3431 | 17.4056 | 18.0057 | 17.3498 | 17.3434 | 17.3429 | 17.343 | 17.3429 |
| | Average | **17.3429** | 17.3532 | 17.3548 | 17.3452 | 17.6932 | 19.0495 | 17.3612 | 17.355 | 17.3515 | 17.3432 | **17.3429** |
| | STD | 3.34E-10 | 2.80E-02 | 0.062361 | 0.0024684 | 0.51805 | 0.49269 | 0.0074302 | 0.059804 | 0.046992 | 0.00014776 | 1.11E-06 |
| F3 | Best | 12.7024 | 12.7024 | 12.7024 | 12.7024 | 12.7025 | 12.7024 | 12.7024 | 12.7024 | 12.7024 | 12.7024 | 12.7024 |
| | Average | **12.7024** | 12.7026 | **12.7024** | **12.7024** | 12.7029 | 12.7027 | **12.7024** | **12.7024** | **12.7024** | **12.7024** | **12.7024** |
| | STD | 6.47E-11 | 6.63E-04 | 8.61E-06 | 7.19E-08 | 0.00028316 | 0.00056048 | 2.11E-06 | 2.52E-06 | 3.04E-06 | 5.14E-09 | 1.33E-10 |
| F4 | Best | 15.9193 | 223.9697 | 7.0437 | 85.4891 | 5439.1073 | 4395.9408 | 99.1087 | 32.6535 | 25.0583 | 135.3144 | 10.3935 |
| | Average | **37.1655** | 1338.3753 | 51.2703 | 236.3547 | 22149.3399 | 9803.3908 | 271.5356 | 882.6031 | 291.8899 | 906.4137 | 44.6316 |
| | STD | 14.4161 | 1371.0254 | 27.0909 | 110.7375 | 7945.0374 | 3958.3639 | 151.0131 | 1224.2067 | 543.0006 | 586.3373 | 22.557 |
| F5 | Best | 1.0492 | 1.0728 | 1.1001 | 1.1478 | 3.0639 | 1.6232 | 1.2012 | 1.1773 | 1.1277 | 2.2907 | 1.0353 |
| | Average | **1.1944** | 1.4568 | 1.3408 | 1.7203 | 4.9309 | 3.2527 | 1.3938 | 1.5554 | 1.3485 | 5.4956 | 1.2888 |
| | STD | 0.13153 | 0.26678 | 0.176 | 0.31529 | 1.4164 | 0.83889 | 0.11893 | 0.24531 | 0.20738 | 1.5584 | 0.23336 |
| F6 | Best | 1.311 | 4.1329 | 6.7499 | 6.8019 | 9.9307 | 7.1442 | 7.4101 | 8.8128 | 4.2781 | 1.543 | 2.6242 |
| | Average | **2.6997** | 7.4505 | 10.2321 | 8.8065 | 12.347 | 8.3606 | 10.1199 | 10.355 | 6.5189 | 3.0625 | 4.8336 |
| | STD | 0.91444 | 1.3591 | 0.89485 | 1.165 | 1.0574 | 0.77485 | 1.0372 | 0.74855 | 1.1973 | 1.1431 | 1.2072 |
| F7 | Best | -138.7496 | -39.9333 | 31.2588 | -31.2209 | 1241.7379 | -207.5513 | -6.7696 | 34.16 | -9.8254 | -107.8243 | -165.2906 |
| | Average | **61.1248** | 310.307 | 302.1934 | 409.561 | 1841.1291 | 140.0036 | 318.3563 | 502.1856 | 286.5727 | 280.5397 | 239.2809 |
| | STD | 107.9929 | 185.5633 | 242.2959 | 251.1559 | 298.9565 | 132.4586 | 202.5213 | 260.6859 | 168.7009 | 149.887 | 169.5606 |
| F8 | Best | 4.0142 | 3.6546 | 2.5078 | 4.6604 | 5.8163 | 3.9991 | 3.9658 | 3.7122 | 3.7871 | 4.3327 | 3.407 |
| | Average | 5.2083 | 5.3185 | **4.7444** | 5.8713 | 6.3994 | 5.2871 | 4.9956 | 5.0948 | 5.2435 | 5.7076 | 5.0643 |
| | STD | 0.57101 | 0.59721 | 1.079 | 0.50496 | 0.32563 | 0.56371 | 0.60652 | 0.68983 | 0.61792 | 0.47102 | 0.62748 |
| F9 | Best | 2.3743 | 2.6131 | 2.6029 | 3.2126 | 1577.7443 | 5.1577 | 2.8024 | 4.0135 | 2.9776 | 2.3394 | 2.3535 |
| | Average | 2.4922 | 16.9539 | 4.1136 | 4.4029 | 3254.5616 | 269.0299 | 4.0988 | 44.7147 | 4.8082 | **2.346** | 2.4403 |
| | STD | 0.078135 | 40.4836 | 0.83175 | 0.71466 | 971.0907 | 210.428 | 0.69102 | 102.9984 | 1.0245 | 0.0045104 | 0.058917 |
| F10 | Best | 1.6462 | 7.8285 | 2.6283 | 20.0194 | 19.9213 | 18.9383 | 2.5855 | 10.1462 | 14.0727 | 19.9836 | 2.00E+01 |
| | Average | 18.7909 | 19.6513 | 19.5043 | 20.1649 | 20.3721 | 20.0392 | **16.2697** | 19.9837 | 19.8815 | 19.9893 | 20.0168 |
| | STD | 4.6108 | 2.2352 | 3.6202 | 0.090939 | 0.12196 | 0.21956 | 7.4463 | 1.9189 | 1.0989 | 0.0061201 | 0.034126 |
| Mean | Rank | 1.8 | 7.3 | 4.6 | 6.9 | 10.3 | 8.1 | 4.7 | 6.7 | 4.6 | 4.7 | 3.4 |
| Final | Ranking | 1 | 9 | 3 | 8 | 11 | 10 | 4 | 7 | 3 | 4 | 2 |

**Table 11**
The Wilcoxon rank sum results between BSLO and other ten algorithms for CEC 2019 functions.

| F | PSO p (Sig.) | GWO p (Sig.) | WOA p (Sig.) | SHO p (Sig.) | AOA p (Sig.) | AO p (Sig.) | GJO p (Sig.) | SCSO p (Sig.) | FOX p (Sig.) | DO p (Sig.) |
|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 2.67E-09 (+) | 4.21E-02 (+) | 1.78E-04 (+) | 0.041609 (-) | 3.02E-11 (+) |
| F2 | 1.85E-01 (~) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) |
| F3 | 2.14E-05 (+) | 8.93E-12 (+) | 1.69E-11 (+) | 5.22E-12 (+) | 5.22E-12 (+) | 5.22E-12 (+) | 5.22E-12 (+) | 8.23E-09 (+) | 5.22E-12 (+) | 2.96E-06 (+) |
| F4 | 3.69E-11 (+) | 8.19E-01 (+) | 4.98E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 9.92E-11 (+) | 6.05E-07 (+) | 3.50E-09 (+) | 3.69E-11 (+) | 3.15E-02 (+) |
| F5 | 5.60E-07 (+) | 1.99E-02 (+) | 2.03E-09 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 1.68E-04 (+) | 8.20E-07 (+) | 1.95E-03 (+) | 3.02E-11 (+) | 2.24E-02 (+) |
| F6 | 3.34E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 6.07E-11 (+) | 4.43E-03 (+) | 4.64E-05 (+) |
| F7 | 1.53E-05 (+) | 2.96E-05 (+) | 1.55E-09 (+) | 3.02E-11 (+) | 1.60E-03 (+) | 1.49E-04 (+) | 1.85E-08 (+) | 8.66E-05 (+) | 3.09E-06 (+) | 6.67E-03 (+) |
| F8 | 1.30E-01 (~) | 1.91E-01 (-) | 6.91E-04 (+) | 8.99E-11 (+) | 1.44E-02 (+) | 1.86E-01 (~) | 6.95E-01 (~) | 5.94E-02 (~) | 1.17E-04 (+) | 6.57E-02 (~) |
| F9 | 3.69E-11 (+) | 4.08E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 3.69E-11 (+) | 3.02E-11 (+) | 3.02E-11 (+) | 4.98E-11 (-) | 7.73E-02 (~) |
| F10 | 2.88E-06 (+) | 3.02E-11 (+) | 2.44E-09 (+) | 3.02E-11 (+) | 1.96E-10 (+) | 1.11E-06 (-) | 3.02E-11 (+) | 7.20E-05 (+) | 3.59E-05 (+) | 1.30E-03 (+) |
| (W\|L\|T) | (8\|0\|2) | (9\|0\|1) | (10\|0\|0) | (10\|0\|0) | (10\|0\|0) | (8\|1\|1) | (9\|0\|1) | (9\|0\|1) | (8\|2\|0) | (8\|0\|2) |

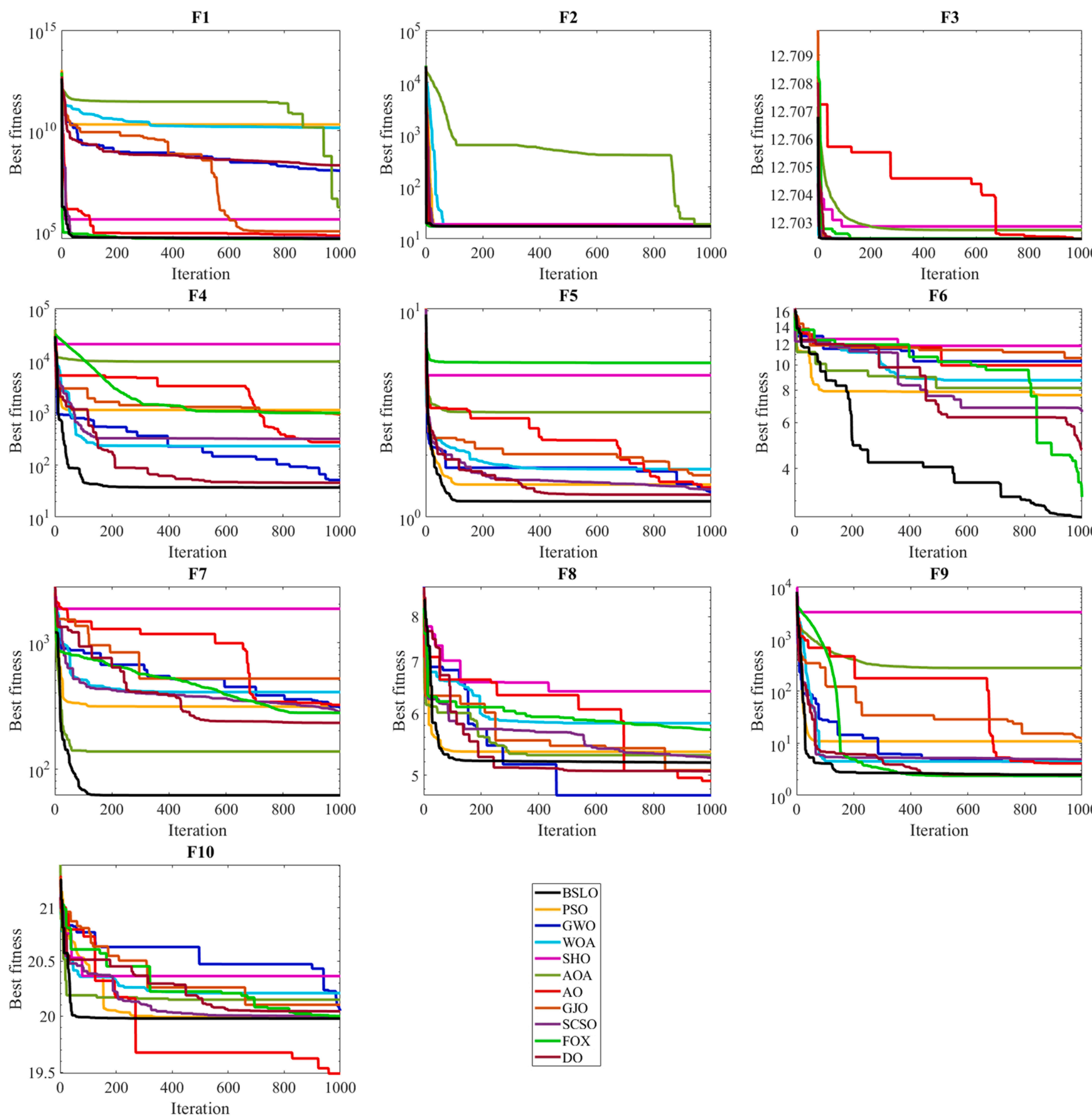


**Fig. 8.** Convergence curves of BSLO and comparative algorithms for CEC 2019 functions.

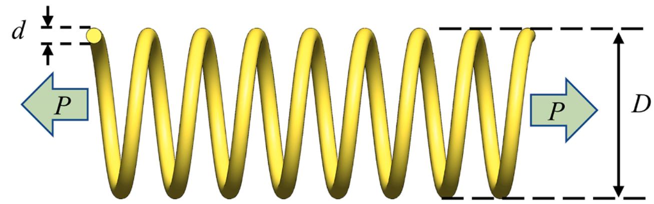


**Fig. 9.** Tension/compression spring design problem.

**Table 12**
Results of BSLO and other ten algorithms for the tension/compression spring design problem.

| Algorithms | Optimum variables | | | Optimum weight ($f(x)$) |
|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | |
| BSLO | 0.051669 | 0.356226 | 11.31788 | **0.0126652** |
| PSO | 0.0516137 | 0.3549069 | 11.3959222 | 0.0126653 |
| GWO | 0.051312 | 0.347702 | 11.83985 | 0.0126699 |
| WOA | 0.051685 | 0.356625 | 11.29444 | 0.0126653 |
| SHO | 0.05 | 0.311194 | 15 | 0.0132258 |
| AOA | 0.05 | 0.316951 | 14.10699 | 0.0127628 |
| AO | 0.051775 | 0.35589 | 11.51831 | 0.0128969 |
| GJO | 0.050844 | 0.336716 | 12.56859 | 0.0126813 |
| SCSO | 0.051686 | 0.356634 | 11.29627 | 0.0126678 |
| FOX | 0.051983 | 0.363808 | 10.88657 | 0.0126686 |
| DO | 0.051673 | 0.356329 | 11.31187 | 0.0126654 |

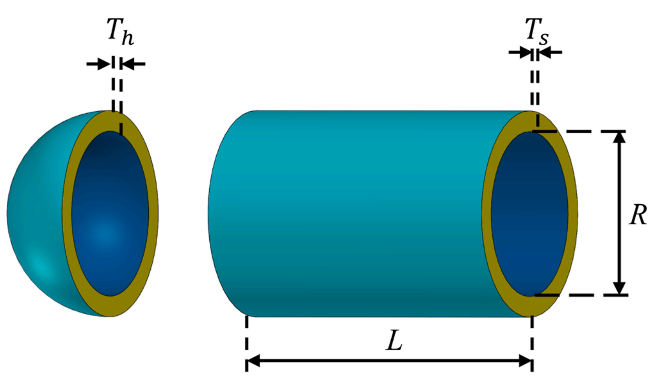


**Fig. 10.** Pressure vessel design problem.

**Table 13**
Results of BSLO and other ten algorithms for the pressure vessel design problem.

| Algorithms | Optimum variables | | | | Optimum cost ($f(x)$) |
|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | |
| BSLO | 0.778169 | 0.3846492 | 40.31962 | 200 | **5885.3328** |
| PSO | 0.8117155 | 0.4012314 | 42.0578 | 177.1409 | 5945.183 |
| GWO | 0.779499 | 0.3860081 | 40.38117 | 199.1471 | 5890.6803 |
| WOA | 0.792683 | 0.4122291 | 41.03167 | 190.3197 | 5976.9524 |
| SHO | 1.51554 | 0.685115 | 69.8617 | 11.6166 | 9979.228 |
| AOA | 0.840572 | 0.4362389 | 42.06167 | 200 | 6810.4485 |
| AO | 0.790223 | 0.4183929 | 40.72199 | 196.1054 | 6053.5901 |
| GJO | 0.778639 | 0.3848958 | 40.33659 | 199.8364 | 5888.7281 |
| SCSO | 0.778182 | 0.3847423 | 40.32019 | 200 | 5885.7933 |
| FOX | 0.780559 | 0.3860622 | 40.43913 | 198.5199 | 5894.5033 |
| DO | 0.778181 | 0.3846643 | 40.32028 | 199.9926 | 5885.418 |

$$x_1 = h,\ x_2 = l,\ x_3 = t,\ x_4 = b$$

Minimize:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

Subject to:

$$g_1(x) = \tau(x) - \tau_{max} \le 0,$$

$$g_2(x) = \sigma(x) - \sigma_{max} \le 0,$$

$$g_3(x) = x_1 - x_4 \le 0,$$

$$g_4(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0,$$

$$g_5(x) = 0.125 - x_1 \le 0,$$

$$g_6(x) = \delta(x) - \delta_{max} \le 0,$$

$$g_7(x) = P - P_c(x) \le 0,$$

Where $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{P}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$, $M = P\left(L + \frac{x_2}{2}\right)$, $R = \sqrt{\frac{x_2^2}{4} + \frac{(x_1+x_3)^2}{2}}$, $J = 2\left[\sqrt{2}x_1x_2\left\{\frac{x_2^2}{4}\right\} + \left(\frac{x_1+x_3}{2}\right)^2\right]$, $\sigma(x) = \frac{6PL}{x_4x_3^2}$, $\delta(x) = \frac{4PL}{Ex_3^3x_4}$, $P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$, $P = 6000\ lb$, $L = 14\ in$, $E = 30 \times 1^6\ psi$, $G = 12 \times 10^6\ psi$, $\tau_{max} = 13600\ psi$, $\sigma_{max} = 30000\ psi$, $\delta_{max} = 0.25\ in$

Variables range:

$$0.1 \le x_1 \le 2,\ 0.1 \le x_2,\ x_3 \le 10,\ 0.1 \le x_4 \le 2$$

Table 14 presents the results of BSLO and comparative algorithms. The best optimum cost is obtained by BSLO, which is 1.72485. The designed variables of $x_1$, $x_2$, $x_3$ and $x_4$ are equal to 0.20573, 3.47049, 9.03662 and 0.20573, respectively. Similarly, BSLO is superior to all comparative algorithms, especially for WOA, SHO, AOA and AO.

### 4.4. Speed reducer design problem

A more complex problem is presented in Fig. 12, called Speed reducer design problem, which possesses eleven constraints and seven design variables [69]. The seven variables are the width of flat ground ($b$), gear module ($m$), the number of teeth of pinion ($p$), the length of the first shaft between bearings ($l_1$), the length of the second shaft between bearings ($l_2$), the diameter of the first bearing ($d_1$), and the diameter of the second bearing ($d_2$). BSLO and other algorithms are considered to optimize the seven variables and search for the optimum weight under the conditions of eleven constraints. The involved mathematical formulation is given as follows.

Seven variables:

$$x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$$

Minimize:

$$f(x) = 0.7854x_1x_2^2 \times \left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$$

Subject to:

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0,$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$

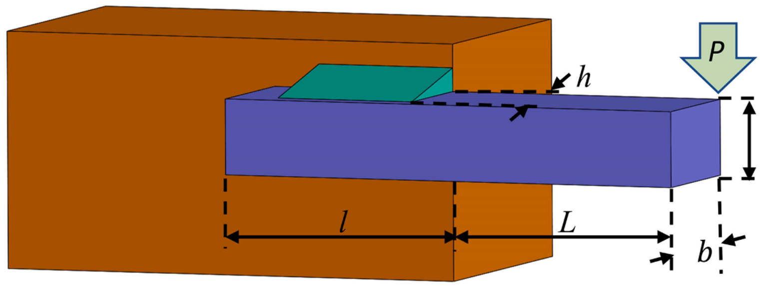


**Fig. 11.** Welded beam design problem.

**Table 14**
Results of BSLO and other ten algorithms for the welded beam design problem.

| Algorithms | Optimum variables | | | | Optimum cost ($f(x)$) |
|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | |
| BSLO | 0.20573 | 3.47049 | 9.03662 | 0.20573 | **1.72485** |
| PSO | 0.20517 | 3.49484 | 9.00537 | 0.20716 | 1.73271 |
| GWO | 0.20544 | 3.47819 | 9.03649 | 0.20577 | 1.72571 |
| WOA | 0.19906 | 3.56105 | 9.24712 | 0.2047 | 1.75513 |
| SHO | 0.23178 | 4.07919 | 8.94753 | 0.23891 | 2.10136 |
| AOA | 0.20162 | 3.50257 | 9.62311 | 0.20572 | 1.82429 |
| AO | 0.19717 | 3.54354 | 9.3936 | 0.20441 | 1.7728 |
| GJO | 0.20564 | 3.46916 | 9.04774 | 0.20572 | 1.7264 |
| SCSO | 0.20566 | 3.47202 | 9.03732 | 0.20573 | 1.72509 |
| FOX | 0.20696 | 3.4519 | 9.01778 | 0.20711 | 1.73142 |
| DO | 0.20573 | 3.47059 | 9.03664 | 0.20573 | 1.72486 |

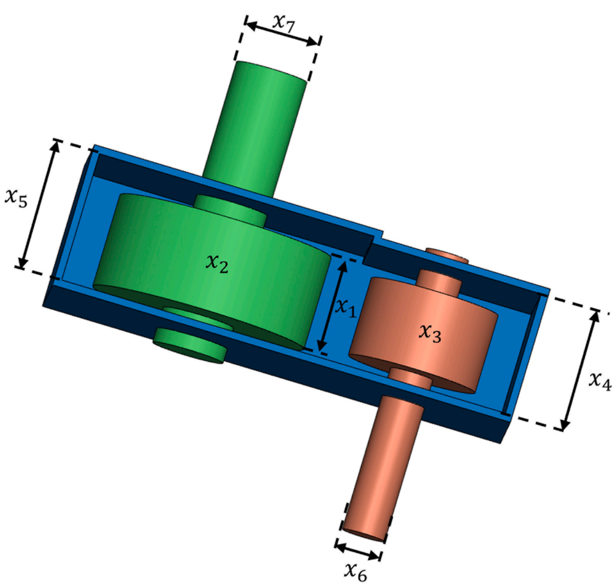


**Fig. 12.** Speed reducer design problem.

$$g_5(x) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \le 0,$$

$$g_6(x) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \le 0,$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0,$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0,$$

$$g_9(x) = \frac{x_2}{12x_2} - 1 \le 0,$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0,$$

Variables range:

$$2.6 \le x_1 \le 3.6,\ 0.7 \le x_2 \le 0.8,\ 17 \le x_3 \le 28,\ 7.3 \le x_4,\ x_5 \le 8.3,\ 2.9 \le x_6 \le 3.9,\ 5 \le x_7 \le 5.5$$

Table 15 presents the optimization results of BSLO and the ten algorithms. When the design variables of $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, $x_6$ and $x_7$ are equal to 3.5, 0.7, 17, 7.3, 7.71532, 3.35021 and 5.28665, respectively, the optimum weight obtained by BSLO is equal to 2994.4711. Results also demonstrate that BSLO wins the ten comparative algorithms for this design problem, especially for GWO, WOA, SHO, AOA, AO, and GJO.

### 4.5. Three-bar truss design problem

The three-bar truss design problem shown in Fig. 13 is a classical problem in civil engineering [49]. This engineering problem aims to look for the minimum weight. This problem with three constraints includes two cross-sectional areas: $A_1(x_1)$ and $A_2(x_2)$. The involved mathematical formulation is given as follows:

Two variables:

$x_1,\ x_2$

Minimize:

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

Subject to:

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

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

**Table 15**
Results of BSLO and other ten algorithms for the speed reducer design problem.

| Algorithms | Optimum variables | | | | | | | Optimum weight ($f(x)$) |
|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | |
| BSLO | 3.5 | 0.7 | 17 | 7.3 | 7.71532 | 3.35021 | 5.28665 | **2994.4711** |
| PSO | 3.50099 | 0.7 | 17 | 7.55694 | 7.82589 | 3.3507 | 5.28669 | 2999.699 |
| GWO | 3.50018 | 0.7 | 17.0022 | 7.33884 | 7.80215 | 3.35524 | 5.2867 | 2998.4902 |
| WOA | 3.51214 | 0.7 | 17 | 7.3 | 7.72315 | 3.35021 | 5.28758 | 3000.0017 |
| SHO | 3.54401 | 0.7 | 17 | 7.3 | 8.3 | 3.59096 | 5.34506 | 3128.0704 |
| AOA | 3.6 | 0.7 | 17 | 7.3 | 8.3 | 3.35216 | 5.29326 | 3051.301 |
| AO | 3.50906 | 0.7 | 17 | 7.66846 | 8.07542 | 3.41234 | 5.30003 | 3033.9799 |
| GJO | 3.50463 | 0.7 | 17 | 7.47823 | 7.86932 | 3.35795 | 5.28776 | 3003.9223 |
| SCSO | 3.50149 | 0.7 | 17.0001 | 7.32868 | 7.73532 | 3.3503 | 5.28698 | 2995.9975 |
| FOX | 3.50025 | 0.7 | 17.0005 | 7.30189 | 7.73502 | 3.35078 | 5.28691 | 2995.4098 |
| DO | 3.5001 | 0.7 | 17 | 7.30002 | 7.71587 | 3.35023 | 5.28666 | 2994.525 |

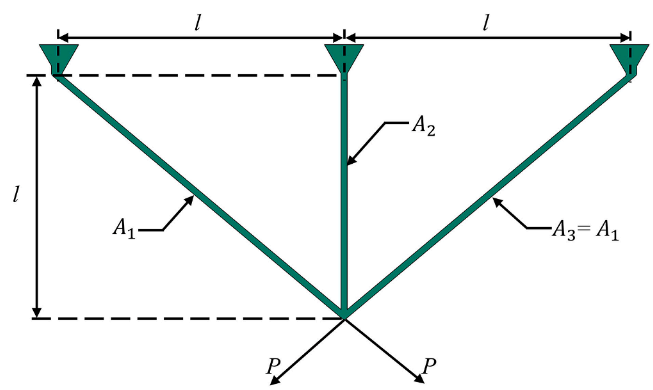


**Fig. 13.** Three-bar truss design problem.

**Table 16**
Results of BSLO and other ten algorithms for the three-bar truss design problem.

| Algorithms | Optimum variables | | Optimum weight ($f(x)$) |
|---|---|---|---|
| | $x_1$ | $x_2$ | |
| BSLO | 0.78867930 | 0.40823651 | **263.8958434** |
| PSO | 0.78867302 | 0.40825427 | **263.8958434** |
| GWO | 0.78866434 | 0.4082794 | 263.8959053 |
| WOA | 0.78867229 | 0.4082564 | 263.8958435 |
| SHO | 0.79442334 | 0.3925893 | 263.9557797 |
| AOA | 0.78525403 | 0.4180925 | 263.9126334 |
| AO | 0.78947013 | 0.4060206 | 263.897935 |
| GJO | 0.78816714 | 0.4096876 | 263.8960955 |
| SCSO | 0.78863367 | 0.4083658 | 263.8958617 |
| FOX | 0.78870269 | 0.4081704 | 263.8958523 |
| DO | 0.78862901 | 0.4083788 | 263.8958452 |

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

Where $l = 100cm$, $P = 2kN/cm^2$,$\sigma = 2kN/cm^2$
Variables range:

$$0 \le x_1,\ x_2 \le 1$$

Table 16 presents the optimization results of BSLO and other ten algorithms. It can be concluded that BSLO with the lowest optimum weight of 263.8958434 is ranked the first compared with other algorithms except for PSO with the same optimum weight. The corresponding $x_1$ and $x_2$ obtained by BSLO are 0.78867930 and 0.40823651. Hence, BSLO is also a better choice for real-world problems than other ten algorithms.

In addition to evaluating optimization performance of BSLO, the

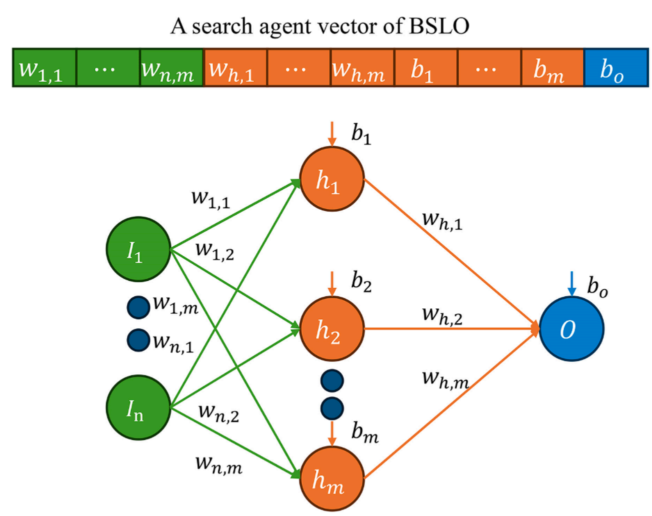


**Fig. 15.** The BSLO-based ANN predictive model.

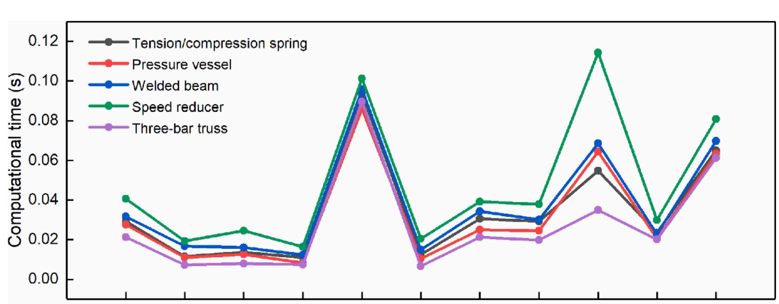


**Fig. 14.** The computational time of BSLO and other ten algorithms for five engineering problems.

**Table 17**
Training dataset for melt electrospinning writing.

| Scenarios | Temperature (°C) | Collector speed (*mm/s*) | Distance (*mm*) | Flow rate (μl/*h*) | Actual Fiber diameter (*μm*) |
|---|---|---|---|---|---|
| 1 | 89.4 | 34.3 | 6.1 | 38.2 | 16.26 |
| 2 | 88 | 38 | 8.2 | 33 | 13.7 |
| 3 | 79.8 | 27.6 | 9.8 | 30.7 | 14.66 |
| 4 | 92.4 | 40.2 | 7.7 | 38.9 | 15.7 |
| 5 | 82 | 25.4 | 8.4 | 25.6 | 13.05 |
| 6 | 77.6 | 32 | 6.7 | 35.9 | 15.74 |
| 7 | 81.3 | 26.9 | 9 | 35.2 | 15.87 |
| 8 | 80.6 | 28.3 | 7.3 | 27.8 | 13.53 |
| 9 | 90.2 | 43.2 | 8.3 | 31.5 | 13.32 |
| 10 | 91.7 | 39.4 | 8.7 | 21.9 | 11.87 |
| 11 | 78.3 | 31.3 | 9.5 | 21.1 | 12.02 |
| 12 | 93.9 | 43.9 | 9.9 | 30 | 13.58 |
| 13 | 94.6 | 26.1 | 8.6 | 36.7 | 19.78 |
| 14 | 79.1 | 35 | 6.4 | 39.6 | 15.7 |
| 15 | 86.5 | 32.8 | 6.2 | 22.6 | 12.18 |
| 16 | 76.1 | 42.4 | 7.4 | 32.2 | 13.55 |
| 17 | 76.9 | 44.6 | 9.3 | 28.5 | 12.17 |
| 18 | 88.7 | 29.8 | 7.1 | 20.4 | 12.49 |
| 19 | 87.2 | 33.5 | 9.2 | 27 | 12.71 |
| 20 | 93.2 | 36.5 | 6.8 | 23.3 | 13.53 |
| 21 | 83.5 | 35.7 | 9.6 | 37.4 | 14.82 |
| 22 | 75.4 | 40.9 | 7.6 | 26.3 | 13.04 |
| 23 | 85 | 41.7 | 6.5 | 29.3 | 12.02 |
| 24 | 90.9 | 30.6 | 7 | 24.1 | 13.68 |
| 25 | 84.3 | 38.7 | 8.9 | 33.7 | 13.09 |
| 26 | 82.8 | 37.2 | 8 | 24.8 | 11.53 |
| 27 | 85.7 | 29.1 | 7.9 | 34.4 | 15.27 |

**Table 18**
Optimization results of BSLO and other ten algorithms for melt electrospinning writing.

| Algorithms | MSE (Ave) | MSE (Min) | MSE (Max) | MSE (STD) |
|---|---|---|---|---|
| BSLO | **0.216** | **0.105** | **0.355** | **0.054** |
| PSO | 0.726 | 0.321 | 1.350 | 0.263 |
| GWO | 0.425 | 0.252 | 0.828 | 0.170 |
| WOA | 1.200 | 0.335 | 2.417 | 0.502 |
| SHO | 1.842 | 0.988 | 5.750 | 1.090 |
| AOA | 1.531 | 0.508 | 3.038 | 0.668 |
| AO | 0.683 | 0.328 | 1.001 | 0.249 |
| GJO | 0.399 | 0.265 | 0.828 | 0.155 |
| SCSO | 0.511 | 0.289 | 0.984 | 0.245 |
| FOX | 0.465 | 0.261 | 0.982 | 0.235 |
| DO | 0.317 | 0.225 | 0.824 | 0.136 |

computational time is also a main factor that needs to be considered for real problems. A good optimization algorithm includes an acceptable computational cost. The computational time of BSLO and other algorithms for five engineering design problems is shown in Fig. 14. It can be seen that the computational time of BSLO is superior to SHO, SCSO, and DO, similar to AO, GJO, and FOX, and only slightly higher than PSO, GWO, WOA, and AOA. However, the performance of PSO, GWO, WOA, and AOA on engineering problems is significantly lower than that of BSLO. BSLO can obtain best optimization results while ensuring good computational efficiency, and is the preferred algorithm compared to others.

## 5. The BSLO-based ANN predictive model for melt electrospinning writing

Melt electrospinning writing is an emerging micro-nano fibre additive manufacturing technology, which has been widely applied for tissue engineering scaffolds and drug delivery [87,88]. The size of fibre diameter affects cell growth and drug delivery rate [104,105]. The manufacture of fibres is a complex process in which the size of fibre diameter is affected by many parameters, mainly including the polymer temperature, the collector speed, tip-to-collector distance, and flow rate [106]. In this study, a single-layer ANN predictive model based on BSLO is proposed to predict the fibre diameter of melt electrospinning writing, as shown in Fig. 15. The ANN model includes four input neurons representing the polymer temperature, the collector speed, tip-to-collector distance, and flow rate, one hidden layer, and an output neuron representing the fibre diameter. The size of neurons in the hidden layer in this study is set to 10. In this ANN model, BSLO is applied to optimize the weights and biases of ANN, and the search agents of BSLO are composed of these variables. The typical sigmoid function is selected as the activation function [107–110]. The Mean Square Error (MSE) between actual and predictive fibre diameters is used to evaluate the fitness values of search agents. The optimization results of BSLO are still compared with that of PSO, GWO, WOA, SHO, AOA, AO, GJO, SCSO, FOX, and DO. The size of search agents and the maximum number of iterations are set to 300 and 1000, respectively. The dimension of this problem with variables taking values [-4, 4] is 61. Each predictive model is run 30 times to ensure the experimental reliability. Here, 27 sets of training data [106] are selected, as shown in Table 17. Four input printing parameters take the values from 75.4~94.6 °C, 25.4~44.6 mm/s, 6.1~9.9 mm, and 20.4~39.6 μl/h. The training process of the BSLO-based ANN predictive model is as follows:

1. The number of ANN neurons and training dataset are defined.
2. The parameters and random candidate solutions of BSLO are initialized.
3. The best fitness value is obtained by calculating the MSE of all search agents.
4. The new weights and biases are generated by the position update functions of BSLO.
5. Step 3-4 are repeated until the end of iteration.
6. The weights and biases with minimal MSE are determined.

Table 18 shows the optimization results of BSLO and comparative algorithms, including average MSE, minimum MSE, maximum MSE, and STD of MSE. The results show that BSLO obtains the best outcome with average MSE = 0.216, minimum MSE = 0.105, maximum MSE = 0.355, and STD of MSE = 0.054. It can be observed that average MSE and STD of MSE obtained by BSLO are lower than those obtained using other algorithms, which means that BSLO owns more stable performance. To intuitively analyse the performance of the ANN predictive model using BSLO and other algorithms, the best training regression curves and convergence curves of all algorithms are shown in Fig. 16 and Fig. 17, respectively. The results of regression curves show that BSLO obtains the best fitting effect with the biggest correlation coefficient $R = 0.98349$, and is obviously superior to other ten algorithms, especially for SHO and AOA. In addition, Fig. 17 shows that BSLO also has the fastest convergence speed compared to other algorithms and can converge to a very small value in the early iterations. Therefore, BSLO obviously outperform other algorithms in terms of fitting training data for the diameter prediction of melt electrospinning writing.

To validate the accuracy of proposed ANN predictive model, six scenarios with random parameter setting are used for the test dataset, as shown in Table 19. Three evaluation criteria including Root-Mean-Square Error (RMSE), Maximum Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are performed to assess the predictive accuracy of the proposed ANN model. The corresponding equations can be found in Refs. [106,111]. Lower values of these three criteria indicate higher predictive accuracy of the proposed predictive model. The results of the predicted fibre diameter and the three criteria are shown in Table 20. The results of Kriging model and the response surface model in Refs [106] are also used for the evaluation of BSLO-based ANN predictive model, as shown in Table 20. The results show that BSLO obtains the best result with RMSE = 0.31, MAE = 0.6 and MAPE = 1.9%, significantly better than other optimization algorithms, the Kriging model, and the response surface model. In addition, the regression curves of test

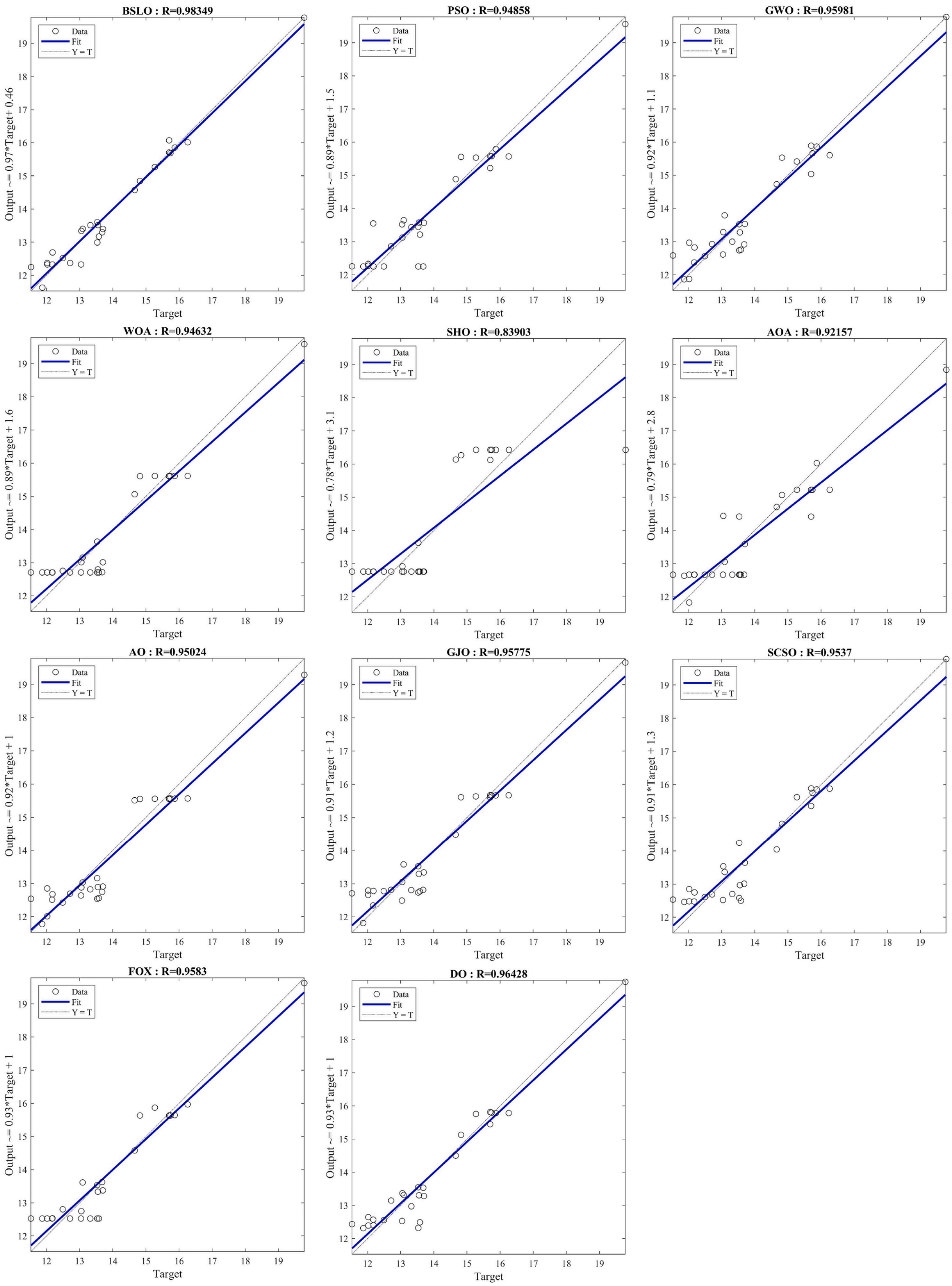


**Fig. 16.** Regression curves of training dataset using BSLO and other ten algorithms.

dataset shown in Fig. 18 also demonstrate that BSLO obtains the best fitting effect with the biggest correlation coefficient R = 0.99053, and obviously outperform other algorithms. Hence, the BSLO-based ANN model is a powerful predictive model and can provide a more accurate predictive method for manufacturing melt electrospinning writing fibres.

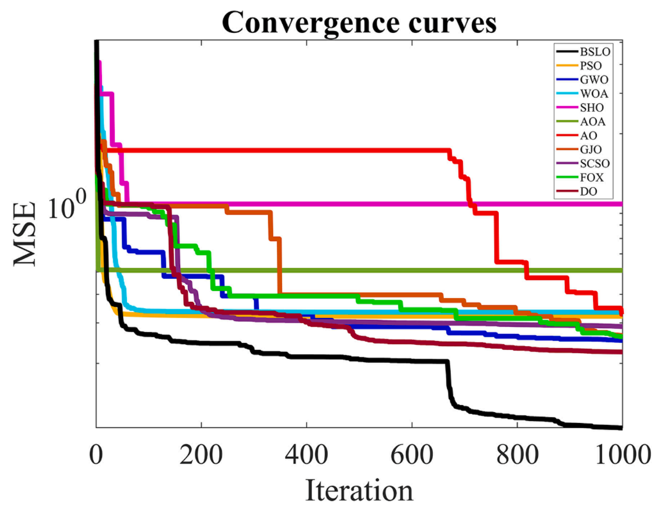


**Fig. 17.** Convergence curves using BSLO and other ten algorithms.

**Table 19**
Test dataset for melt electrospinning writing.

| Scenarios | Temperature (°C) | Collector speed (*mm/s*) | Distance (*mm*) | Flow rate ($\mu l/h$) | Actual Fiber diameter ($\mu m$) |
|---|---|---|---|---|---|
| 1 | 85 | 35 | 10 | 20 | 10.5 |
| 2 | 85 | 45 | 6 | 30 | 12.45 |
| 3 | 95 | 25 | 10 | 30 | 17.34 |
| 4 | 95 | 45 | 8 | 20 | 12.37 |
| 5 | 75 | 45 | 6 | 40 | 14.58 |
| 6 | 75 | 35 | 8 | 30 | 13.65 |

## 6. Conclusions and future work

This study introduces a novel meta-heuristic optimization algorithm, called Blood-Sucking Leech Optimizer (BSLO), which is inspired by the foraging behavior of the blood-sucking leech in rice fields. The mathematical model of BSLO is developed by five strategies, including the exploration of directional leeches, exploitation of directional leeches, switching mechanism of directional leeches, search strategy of directionless leeches, and re-tracking strategy.

The optimization performance of BSLO is assessed using twenty-three classical benchmark functions. Parameter analysis results show that BSLO with parameter setting $(m, a, b, b_2, t_1, t_2) = (0.8, 0.97, 0.001, 0.00001, 20, 20)$ can obtain the best optimization performance. Four qualitative analyses including the search history, the 1D trajectory, the average fitness values and the convergence curve show that BSLO possesses strong abilities of exploration, exploitation, and local optima avoidance, as well as fast convergence speed. The results calculated by BSLO are compared with those obtained by ten meta-heuristic algorithms, including PSO, GWO, WOA, SHO, AOA, AO, GJO, SCSO, FOX, and DO. The Friedman ranking and Wilcoxon signed-rank test results show that BSLO wins against all comparative algorithms. The results of unimodal functions indicated that BSLO possesses better exploitation ability than the comparative algorithms. The results of BSLO for multimodal functions demonstrated its superiority in terms of the abilities of exploitation and local optima avoidance. Furthermore, the convergence curves indicated that BSLO has a very fast convergence speed for almost all 23 classical benchmark functions and outperformed the comparative algorithms. The stability analysis showed that BSLO owns the most stable performance for unimodal functions and multimodal functions with 30, 100, and 500 dimensions. To further assess the efficiency of BSLO for solving complex problems, CEC 2017 and 2019 test functions are selected. Numerical experiment results show that BSLO still remains the strongest abilities of exploration, exploitation, and local optima avoidance, as well as fastest convergence speed.

To research BSLO's applicability for real-world problems, several classical engineering design problems are considered, namely tension/compression spring design problem, pressure vessel design problem, welded beam design problem, speed reducer design problem, and three-bar truss design problem. The optimization results show that BSLO obtains the lowest weight or cost for five engineering design problems than the other ten algorithms. This indicates that BSLO still owns very strong optimization performance for solving these problems with multiple constraints. In addition, the BSLO's applicability is also demonstrated using a BSLO-based ANN predictive model for melt electrospinning writing fibre. The results show that the BSLO-based ANN predictive model obtains the best fitting results for the training, validation and testing dataset, and possesses the best predictive accuracy compared with other predictive models. It also provides a new and powerful method to predict the fibre diameter of melt electrospinning writing. Hence, BSLO is a potential meta-heuristic optimizer for constrained and unconstrained engineering problems in real-world applications.

Overall, BSLO showed a great optimization performance in both numerical and engineering problems owing to good balance strategies between exploration and exploitation, and strong ability of local optima avoidance. However, BSLO's performance was only evaluated for simple-objective problems, and is not investigated on multi-objective problems. Moreover, it is undoubted that how to find a better balance between exploration and exploitation is still a huge challenge. Hence, our future work is to assess BSLO's performance for multi-objective problems and provide a better balance strategy.

**Table 20**
Validation results of BSLO and other algorithms model for melt electrospinning writing.

| Algorithms | Predicted fibre diameter ($\mu m$) | | | | | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|---|---|---|---|
| | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 | Scenario 5 | Scenario 6 | | | |
| BSLO | 11.10 | 12.33 | 17.71 | 12.30 | 14.61 | 13.41 | **0.31** | **0.60** | **1.90%** |
| PSO | 12.25 | 12.26 | 17.49 | 12.25 | 15.57 | 13.65 | 0.83 | 1.75 | 4.47% |
| GWO | 11.71 | 12.95 | 14.12 | 11.70 | 15.39 | 13.53 | 1.48 | 3.22 | 7.65% |
| WOA | 12.71 | 12.71 | 17.50 | 12.71 | 15.62 | 13.04 | 1.04 | 2.21 | 6.38% |
| SHO | 12.76 | 12.76 | 16.43 | 12.76 | 12.83 | 12.76 | 1.29 | 2.26 | 8.48% |
| AOA | 9.06 | 12.66 | 14.05 | 12.57 | 14.41 | 12.80 | 1.51 | 3.29 | 7.22% |
| AO | 11.39 | 12.86 | 15.50 | 11.21 | 15.57 | 12.98 | 1.09 | 1.84 | 7.23% |
| GJO | 11.83 | 12.74 | 17.14 | 9.38 | 15.66 | 13.50 | 1.42 | 2.99 | 8.14% |
| SCSO | 12.46 | 12.78 | 14.46 | 12.45 | 15.38 | 13.24 | 1.47 | 2.88 | 7.84% |
| FOX | 12.53 | 12.53 | 16.48 | 12.53 | 13.33 | 13.03 | 1.07 | 2.03 | 6.54% |
| DO | 12.32 | 12.54 | 19.58 | 12.28 | 13.46 | 13.31 | 1.27 | 2.24 | 6.98% |
| RSM [106] | 10.73 | 11.20 | 18.29 | 12.31 | 14.35 | 14.83 | 0.81 | 1.25 | 4.74% |
| Kriging model [106] | 11.10 | 12.07 | 17.78 | 12.57 | 15.00 | 14.38 | 0.49 | 0.73 | 3.52% |

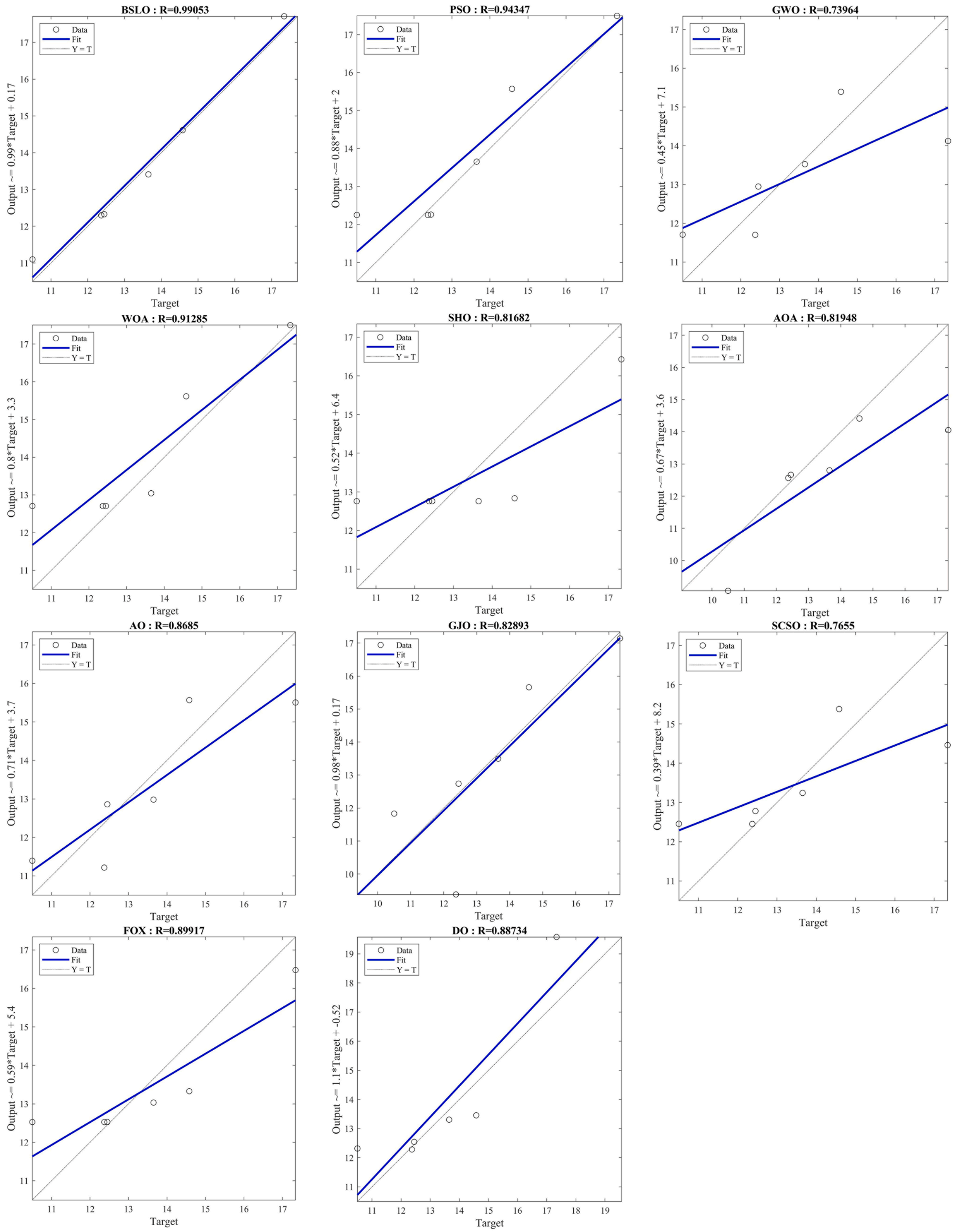


**Fig. 18.** Regression curves of test dataset using BSLO and other algorithms.

## CRediT authorship contribution statement

**Jianfu Bai:** Writing – original draft, Validation, Methodology, Investigation. **H. Nguyen-Xuan:** Writing – review & editing, Supervision. **Elena Atroshchenko:** Writing – review & editing. **Gregor Kosec:** Writing – review & editing. **Lihua Wang:** Supervision. **Magd Abdel Wahab:** Writing – review & editing, Validation, Supervision, Conceptualization.

## 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.

## Data availability

Data will be made available on request.

## Acknowledgments

The study is supported financially by the grants from the China Scholarship Council (202108440192). This work is supported by the National Natural Science Foundation of China (Project no. 12272270) and the Shanghai Pilot Program for Basic Research. G.K. acknowledges the financial support from the Slovenian Research Agency research core funding No. P2-0095.

## References

[1] Zaher H, Al-Wahsh H, Eid MH, Gad RSA, Abdel-Rahim N, Abdelqawee IM. A novel harbor seal whiskers optimization algorithm. Alexandria Eng J 2023;80: 88–109.
[2] Rizk-Allah RM, Saleh O, Hagag EA, Mousa AAA. Enhanced tunicate swarm algorithm for solving large-scale nonlinear optimization problems. Int J Comput Intell Syst 2021;14.
[3] Bai J, Li Y, Zheng M, Khatir S, Benaisa B, Abualigah L, et al. A sinh cosh optimizer. Knowledge-based systems. 2023, 111081.
[4] YiFei L, MaoSen C, Ngoc HT, Khatir S, Abdel Wahab M. Multi-parameter identification of concrete dam using polynomial chaos expansion and slime mould algorithm. Comput Struct 2023;281.
[5] Minh H-L, Sang-To T, Abdel Wahab M, Cuong-Le T. A new metaheuristic optimization based on K-means clustering algorithm and its application to structural damage identification. Knowl Based Syst 2022;251.
[6] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press; 1992.
[7] YiFei L, Minh H-L, Khatir S, Sang-To T, Cuong-Le T, MaoSen C, et al. Structure damage identification in dams using sparse polynomial chaos expansion combined with hybrid K-means clustering optimizer and genetic algorithm. Eng Struct 2023;283.
[8] Es-Haghi MS, Shishegaran A, Rabczuk T. Evaluation of a novel Asymmetric Genetic Algorithm to optimize the structural design of 3D regular and irregular steel frames. Front Struct Civil Eng 2020;14:1110–30.
[9] Beyer H-G, Schwefel H-P. Evolution strategies–a comprehensive introduction. Nat Comput 2002;1:3–52.
[10] Juste K, Kita H, Tanaka E, Hasegawa J. An evolutionary programming solution to the unit commitment problem. IEEE Trans Power Syst 1999;14:1452–9.
[11] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optimiz 1997;11:341–59.
[12] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 2013;219:8121–44.
[13] Ghaemi M, Feizi-Derakhshi M-R. Forest optimization algorithm. Expert Syst Appl 2014;41:6676–87.
[14] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. Science (1979) 1983;220:671–80.
[15] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[16] Tahani M, Babayan N. Flow Regime Algorithm (FRA): a physics-based meta-heuristics algorithm. Knowl Inf Syst 2018;60:1001–38.
[17] Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Applied Intell 2020;51:1531–51.
[18] Pereira JLJ, Francisco MB, Diniz CA, Antônio Oliver G, Cunha SS, Gomes GF. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. Expert Syst Appl 2021;170.
[19] Karami H, Anaraki MV, Farzin S, Mirjalili S. Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems. Comput Ind Eng 2021;156.
[20] Hashim FA, Mostafa RR, Hussien AG, Mirjalili S, Sallam KM. Fick's Law Algorithm: A physical law-based algorithm for numerical optimization. Knowl Based Syst 2023;260.
[21] Deng L, Liu S. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. Expert Syst Appl 2023;225.
[22] Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M. Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowl Based Syst 2023;268.
[23] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput-Aided Design 2011;43:303–15.
[24] Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R. The social engineering optimizer (SEO). Eng Appl Artific Intell 2018;72:267–93.
[25] Samareh Moosavi SH, Bardsiri VK. Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intell 2019;86: 165–81.
[26] Askari Q, Saeed M, Younas I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. Expert Syst Appl 2020;161.
[27] Zhang Y, Jin Z. Group teaching optimization algorithm: a novel metaheuristic method for solving global optimization problems. Expert Syst Appl 2020:148.
[28] Naik A, Satapathy SC. Past present future: a new human-based algorithm for stochastic optimization. Soft comput 2021;25:12915–76.
[29] Dehghani M, Trojovska E, Trojovsky P. A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. Sci Rep 2022;12:9924.
[30] Tanhaeean M, Tavakkoli-Moghaddam R, Akbari AH. Boxing Match Algorithm: a new meta-heuristic algorithm. Soft comput 2022;26:13277–99.
[31] Zhang W, Pan K, Li S, Wang Y. Special Forces Algorithm: A novel meta-heuristic method for global optimization. Math Comput Simul 2023;213:394–417.
[32] Hussain K, Mohd Salleh MN, Cheng S, Shi Y. Metaheuristic research: a comprehensive survey. Artif Intell Rev 2018;52:2191–233.
[33] Camacho-Villalón CL, Stützle T, Dorigo M. Designing new metaheuristics: manual versus automatic approaches. Intell Comput 2023;2:0048.
[34] Camacho-Villalón CL, Dorigo M, Stützle T. Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. Int Trans Operat Res 2023;30:2945–71.
[35] Camacho-Villalón CL, Dorigo M, Stützle T. An analysis of why cuckoo search does not bring any novel ideas to optimization. Comput Oper Res 2022;142:105747.
[36] Li C, Amanov A, Li Y, Wang C, Wang D, Wahab MA. Prediction of residual stress distribution induced by ultrasonic nanocrystalline surface modification using machine learning. Adv Eng Soft 2024;188:103570.
[37] Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intell 2007; 1:33–57.
[38] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag 2006;1:28–39.
[39] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optimiz 2007;39: 459–71.
[40] Gad AG. Particle swarm optimization algorithm and its applications: a systematic review. Arch Comput Methods Eng 2022;29:2531–61.
[41] Kaya E, Gorkemli B, Akay B, Karaboga D. A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. Eng Appl Artif Intell 2022;115.
[42] Öztürk Ş, Ahmad R, Akhtar N. Variants of Artificial Bee Colony algorithm and its applications in medical image processing. Appl Soft Comput 2020;97.
[43] Faris H, Aljarah I, Al-Betar MA, Mirjalili S. Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl 2017;30:413–35.
[44] Jain M, Singh V, Rani A. A novel nature-inspired algorithm for optimization: squirrel search algorithm. Swarm Evol Comput 2019;44:148–75.
[45] Khishe M, Mosavi MR. Chimp optimization algorithm. Expert Syst Appl 2020;149.
[46] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. Future Generat Comput Syst 2020;111: 300–23.
[47] Hayyolalam V, Pourhaji Kazem AA. Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 2020;87.
[48] Mohammadi-Balani A, Dehghan Nayeri M, Azar A, Taghizadeh-Yazdi M. Golden eagle optimizer: a nature-inspired metaheuristic algorithm. Comput Ind Eng 2021;152.
[49] Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 2021;158.
[50] Braik M, Hammouri A, Atwan J, Al-Betar MA, Awadallah MA. White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl Based Syst 2022;243.
[51] Hashim FA, Hussien AG. Snake optimizer: a novel meta-heuristic optimization algorithm. Knowl Based Syst 2022;242.
[52] El-kenawy E-SM, Khodadadi N, Mirjalili S, Abdelhamid AA, Eid MM, Ibrahim A. Greylag goose optimization: nature-inspired optimization algorithm. Expert Syst Appl 2024;238.
[53] Mirjalili S. The ant lion optimizer. Adv Eng Soft 2015;83:80–98.
[54] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 2015;89:228–49.
[55] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Soft 2017;114:163–91.
[56] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Adv Eng Soft 2017;105:30–47.
[57] Gomes GF, da Cunha SS, Ancelotti AC. A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Eng Comput 2018;35:619–26.
[58] Zhao W, Zhang Z, Wang L. Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 2020;87.

[59] Le-Duc T, Nguyen Q-H, Nguyen-Xuan H. Balancing composite motion optimization. Inf. Sci. 2020;520:250–70.
[60] Al-Sorori W, Mohsen AM. New Caledonian crow learning algorithm: a new metaheuristic algorithm for solving continuous optimization problems. Appl Soft Comput 2020;92.
[61] Jiang Y, Wu Q, Zhu S, Zhang L. Orca predation algorithm: a novel bio-inspired algorithm for global optimization problems. Expert Syst Appl 2022:188.
[62] Zhao W, Wang L, Mirjalili S. Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 2022:388.
[63] Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. Math Comput Simul 2022;192:84–110.
[64] Połap D, Woźniak M. Red fox optimization algorithm. Expert Syst Appl 2021:166.
[65] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 2022;114.
[66] Pan J-S, Zhang L-G, Wang R-B, Snášel V, Chu S-C. Gannet optimization algorithm: a new metaheuristic algorithm for solving engineering optimization problems. Math Comput Simul 2022;202:343–73.
[67] Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl Based Syst 2023;259:110011.
[68] Zamani H, Nadimi-Shahraki MH, Gandomi AH. Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. Comput Methods Appl Mech Eng 2022;392.
[69] Minh H-L, Sang-To T, Theraulaz G, Abdel Wahab M, Cuong-Le T. Termite life cycle optimizer. Expert Syst Appl 2023;213.
[70] Azizi M, Talatahari S, Gandomi AH. Fire Hawk optimizer: a novel metaheuristic algorithm. Artif Intell Rev 2022;56:287–363.
[71] Xian S, Feng X. Meerkat optimization algorithm: a new meta-heuristic optimization algorithm for solving constrained engineering problems. Expert Syst Appl 2023;231.
[72] Hu G, Guo Y, Wei G, Abualigah L. Genghis Khan shark optimizer: a novel nature-inspired algorithm for engineering optimization. Adv Eng Inf 2023;58.
[73] Han M, Du Z, Yuen KF, Zhu H, Li Y, Yuan Q. Walrus optimizer: a novel nature-inspired metaheuristic algorithm. Expert Syst Appl 2024;239.
[74] Chen H, Xu Y, Wang M, Zhao X. A balanced whale optimization algorithm for constrained engineering design problems. Appl Math Modell 2019;71:45–59.
[75] Zhao S, Zhang T, Ma S, Wang M. Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl Intell 2022;53:11833–60.
[76] Chen H, Li W, Yang X. A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. Expert Syst Appl 2020;158.
[77] Chakraborty S, Saha AK, Chakraborty R, Saha M. An enhanced whale optimization algorithm for large scale optimization problems. Knowl Based Syst 2021;233.
[78] Nadimi-Shahraki MH, Taghian S, Mirjalili S. An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 2021;166.
[79] Yu X, Xu W, Li C. Opposition-based learning grey wolf optimizer for global optimization. Knowl Based Syst 2021;226.
[80] Jiang J, Zhao Z, Liu Y, Li W, Wang H. DSGWO: an improved grey wolf optimizer with diversity enhanced strategy based on group-stage competition and balance mechanisms. Knowl Based Syst 2022;250.
[81] Dong H, Xu Y, Li X, Yang Z, Zou C. An improved antlion optimizer with dynamic random walk and dynamic opposite learning. Knowl Based Syst 2021;216.
[82] Si T, Miranda PBC, Bhattacharya D. Novel enhanced Salp Swarm Algorithms using opposition-based learning schemes for global optimization problems. Expert Syst Appl 2022;207.
[83] Mostafa RR, Gaheen MA, Abd ElAziz M, Al-Betar MA, Ewees AA. An improved gorilla troops optimizer for global optimization problems and feature selection. Knowl Based Syst 2023;269.
[84] Yao L, Yuan P, Tsai C-Y, Zhang T, Lu Y, Ding S. ESO: an enhanced snake optimizer for real-world engineering problems. Expert Syst Appl 2023;230.
[85] Lou T-s, Yue Z-p, Jiao Y-z, He Z-d. A hybrid strategy-based GJO algorithm for robot path planning. Expert Syst Appl 2024;238.
[86] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82.
[87] Bai J, Wang H, Gao W, Liang F, Wang Z, Zhou Y, et al. Melt electrohydrodynamic 3D printed poly (epsilon-caprolactone)/polyethylene glycol/roxithromycin scaffold as a potential anti-infective implant in bone repair. Int J Pharm 2020;576:118941.
[88] Xiong J, Wang H, Lan X, Wang Y, Wang Z, Bai J, et al. Fabrication of bioinspired grid-crimp micropatterns by melt electrospinning writing for bone–ligament interface study. Biofabrication 2022;14:025008.
[89] Abdualkader A, Ghawi A, Alaama M, Awang M, Merzouk A. Leech therapeutic applications. Indian J Pharm Sci 2013;75:127.
[90] Harley CM, Cienfuegos J, Wagenaar DA. Developmentally regulated multisensory integration for prey localization in the medicinal leech. J Experiment Biol 2011;214:3801–7.
[91] Harley CM, Rossi M, Cienfuegos J, Wagenaar D. Discontinuous locomotion and prey sensing in the leech. J Experiment Biol 2013;216:1890–7.
[92] Miao Y, Guo Q, Shi H, Wang J, Guo L. Study on foraging mechanism of leeches with different feeding habits based on chemoreception and foraging behavior. Invertebrate Biol 2023;142.
[93] Dickinson MH, Lent CM. Feeding behavior of the medicinal leech. Hirudo Medicinalis L J Compar Physiol A 1984;154:449–55.
[94] Shaaban AM, Anitescu C, Atroshchenko E, Rabczuk T. An isogeometric Burton-Miller method for the transmission loss optimization with application to mufflers with internal extended tubes. Applied Acoustics 2022;185.
[95] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Soft 2014;69:46–61.
[96] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Soft 2016;95:51–67.
[97] Dhiman G, Kumar V. Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Soft 2017;114:48–70.
[98] Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 2021;376.
[99] Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH. Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 2021;157.
[100] Chopra N, Mohsin Ansari M. Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst Appl 2022;198.
[101] Seyyedabbasi A, Kiani F. Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Eng Comput 2022;39:2627–51.
[102] Mohammed H, Rashid T. FOX: a FOX-inspired optimization algorithm. Appl Intell 2022;53:1030–50.
[103] Zhao S, Zhang T, Ma S, Chen M. Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. Eng Appl Artif Intell 2022;114.
[104] Gao Q, Xie C, Wang P, Xie M, Li H, Sun A, et al. 3D printed multi-scale scaffolds with ultrafine fibers for providing excellent biocompatibility. Mater Sci Eng C Mater Biol Appl 2020;107:110269.
[105] Lan X, Wang H, Bai J, Miao X, Lin Q, Zheng J, et al. Multidrug-loaded electrospun micro/nanofibrous membranes: fabrication strategies, release behaviors and applications in regenerative medicine. J Control Release 2021;330:1264–87.
[106] Xie Y, Chen J, Zhao H, Huang F. Prediction of the fiber diameter of melt electrospinning writing by kriging model. J Appl Polym Sci 2022;139.
[107] Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. Comput Mater Contin 2019;59:345–59.
[108] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. Comput Methods Appl Mech Eng 2020;362:112790.
[109] Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of Kirchhoff plate. arXiv preprint arXiv; 2021. 210202617.
[110] Bai J, Khatir S, Abualigah L, Abdel Wahab M. Ameliorated Golden jackal optimization (AGJO) with enhanced movement and multi-angle position updating strategy for solving engineering problems. Adv Eng Soft 2024;194.
[111] Goswami S, Anitescu C, Chakraborty S, Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theoretic Appl Fracture Mech 2020;106:102447.