Knowledge-Based Systems 282 (2023) 111081

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A Sinh Cosh optimizer

Jianfu Bai [a], Yifei Li [a], Mingpo Zheng [a], Samir Khatir [a], Brahim Benaissa [b], Laith Abualigah [c,d,e,f], Magd Abdel Wahab [a,g,*]

[a] Soete Laboratory, Department of Electrical Energy, Metals, Mechanical Constructions, and Systems, Faculty of Engineering and Architecture, Ghent University, Belgium
[b] Design Engineering Laboratory, Toyota Technological Institute, Nagoya, Japan
[c] Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon
[d] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
[e] MEU Research Unit, Middle East University, Amman, Jordan
[f] Applied science research center, Applied science private university, Amman 11931, Jordan
[g] College of Engineering, Yuan Ze University, Taiwan

A R T I C L E I N F O

Keywords:
Meta-heuristic
Sinh Cosh Optimizer (SCHO)
Mathematical inspiration
Benchmark function
Engineering design problems

A B S T R A C T

Currently, meta-heuristic algorithms have been widely studied and applied, but balancing exploration and exploitation remains a challenge. In this study, a novel meta-heuristic algorithm named Sinh Cosh Optimizer (SCHO) is proposed based on the mathematical inspiration of the characteristics of Sinh and Cosh. SCHO includes four steps: two different phases of exploration and exploitation, the bounded search strategy, and the switching mechanism. SCHO is compared with eight meta-heuristic algorithms for the 23 benchmark functions at different dimensions and CEC 2014, and its strong performance is validated. The efficiency and robustness of SCHO are verified by qualitative analysis, convergence curves, and two statistical tests. Furthermore, five engineering problems are presented. Source codes of SCHO are publicly available at https://www.mathworks.com/matlabcentral/fileexchange/130734-a-sinh-cosh-optimizer.

1. Introduction

With the emergence of more nonlinear, large-scale, high-dimensional, and complex constrained problems, meta-heuristic optimization algorithms have been researched by more and more researchers [1]. Although traditional exact optimization algorithms can obtain more accuracy solutions than meta-heuristic in small-scale problems by searching the whole space, it does not only cost a lot of time, but also it is unrealistic to optimize the engineering problems with large-scale and high-dimension search space [2]. On the contrary, meta-heuristic optimization algorithms as methods of approximation are effective for these complex problems due to their simple principles and their better avoidance of local optimum [3,4]. Moreover, meta-heuristic optimization algorithms do not require derivation for optimization problems and are only concerned with the input and output data. The process can be regarded as a black box, which indicates that meta-heuristic optimization algorithms possess good flexibility and can be used for optimization in various fields [5].

Meta-heuristic optimization algorithms always perform a repetitive procedure of "trial and error", which means that meta-heuristic algorithms can update new solutions at each iteration to find the minimal or maximal values for optimization problems. The new solutions will be compared with the optimal solution obtained so far [6]. Meta-heuristic algorithms can be grouped into two types via the number of candidate solutions and different inspiration methods. According to the first type, meta-heuristic algorithms are divided into single-solution-based and population-based optimization algorithms [7,8]. The single-solution-based algorithms, such as Simulated Annealing (SA) inspired by annealing process of metals, Vortex Search (VS) inspired by the vortex pattern occurred in a stirred liquid [9], Social Engineering Optimizer (SEO) inspired by Social Engineering [10], and single-solution Simulated Kalman Filter (ssSKF) inspired by Kalman Filter [11], only produce one candidate solution at each iteration. Then, the new solution is compared with the optimal solution obtained so far to decide whether the optimal solution is replaced. These algorithms possess short computing time. Still, bad exploration and rapid convergence of these algorithms occur so that these algorithms easily get stuck in a local optimum. By contrast, the population-based algorithms can search the whole space of optimization problems by using many candidate solutions at each iteration to improve the ability of local

* Corresponding author.
E-mail address: magd.abdelwahab@ugent.be (M. Abdel Wahab).

https://doi.org/10.1016/j.knosys.2023.111081
Received 8 April 2023; Received in revised form 23 September 2023; Accepted 13 October 2023
Available online 18 October 2023

optimum avoidance and looking for a global optimum. The disadvantages of population-based algorithms are more function evaluation and longer runtime.

A more common classified method is based on different inspirations, and divides meta-heuristic algorithms into four classes: Evolutionary Algorithm (EA), Physics-Based Algorithm (PhA), Human-Based Algorithm (HBA), and Swarm Intelligence (SI) algorithm [12,13]. EA is proposed based on the inspiration of the principle of natural evolution. Genetic Algorithm (GA), the most famous EA, was proposed by Holland based on the inspiration of Darwinian evolution [14]. GA mainly performs the crossover and mutation strategies to obtain better solutions to replace the poorer ones. The other well-known EA, such as Genetic Programming (GP) [15], Differential Evolution (DE) [16], Evolutionary Programming (EP) [17], Evolution Strategy (ES) [18], are all involved in mutation, crossover and recombination. Biogeography-Based Optimization (BBO) algorithm mainly focuses on migration and exchange of information [19]. PhA is proposed according to the inspiration of natural laws of physics, and Simulated Annealing (SA) belongs to the classical representative of PhA [20]. The performance of SA inspired by the physical concept of annealing is affected by temperature. The probability that the optimal solution obtained so far is replaced by a bad solution gradually decreases by decreasing the temperature until SA converges to the optimal solution or close to it. Also, other examples of PhA include Gravitational Search Algorithm (GSA) inspired by the law of interaction between gravity and mass [21], Charged System Search (CSS) based on Coulomb's law and Newtonian mechanics laws [22], Multi-Verse Optimizer (MVO) based on black hole, white hole, and wormhole in universe [23], Nuclear Reaction Optimization (NRO) inspired by nuclear reaction process [24], Atom Search Optimization (ASO) based on basic molecular dynamics [25], Lichtenberg Algorithm (LA) inspired by Lichtenberg figures patterns [26], Flow Direction Algorithm (FDA) inspired by the direction of flow toward the outlet of a drainage basin [27], Light Spectrum Optimizer (LSO) inspired by light dispersions though raindrops [28], Fick's Law Algorithm (FLA) inspired by Fick's first rule of diffusion [29], Kepler Optimization Algorithm (KOA) based on the inspiration of Kepler's laws of planetary motion [30], and Snow Ablation Optimizer (SAO) inspired by sublimation and melting of snow [31].

Human-based algorithms are proposed based on the inspiration of human production and life behaviors. One of the most well-known algorithms is Teaching-Learning-Based Optimization (TLBO), which searches for the optimal solution by using two steps, the 'Teacher Phase' and the 'Learner Phase' [32–34]. Other examples are League Championship Algorithm (LCA) based on contest between sports teams in a sport league [35], Colliding Bodies Optimization (CBO) inspired by collision between bodies [36], Collective Decision Optimization (CSO) based on five decision-making characteristics of human [37], Poor and Rich Optimization (PRO) based on efforts by the poor and the rich to achieve a better life [38], Gaining Sharing Knowledge (GSK) based Algorithm inspired by knowledge acquisition and sharing behaviors [39], Equilibrium Optimizer (EO) based on control volume mass balance [40], Past Present Future (PPF) inspired by learning from successful people to improve their lives [41], Chef-Based Optimization Algorithm (CBOA) inspired by the process of learning to cook [42], Mountaineering Team-Based Optimization (MTBO) inspired by human cooperative behavior to get the mountain top [43], Human Felicity Algorithm (HFA) inspired by human collective intelligence to obtain felicity for hyper-parameter tuning [44].

The last group of SI algorithms is the most widely researched meta-heuristics [45]. SI algorithms is based on the inspiration of the hunting, multiplication, and hierarchical relationship of animals or plants [46,47]. The optimization process of SI algorithms mainly includes three steps. Firstly, the population candidate solutions are initialized randomly. Then, the next position of candidate solutions is updated mainly using a strategy based on the current position and the optimal solution obtained so far. Finally, the optimal solution at each iteration is selected by comparing the new solutions and optimal solutions obtained so far. The most famous SI algorithm is Particle Swarm Optimization (PSO) [48]. PSO is based on the inspiration of the collective behavior of a school of fish or birds, and searches the search space by continuously updating the new solutions according to their position. The optimal solution is updated until the end of iterations. There are other well-known SI algorithms that are widely studied and used. Ant Colony Optimization (ACO) is a SI algorithm inspired by foraging of ants, which simulates the phenomenon that ants can choose an optimal way according to the amount of pheromones [49]. Bat Algorithm (BA) is based on inspiration of the echolocation of bats, in which bats update their velocities and positions by using three strategies including frequency tuning, velocity updating and echolocation [50]. Cuckoo Search (CS) Algorithm was introduced by Xin-She Yang for solving structural engineering optimization, which simulates obligate nest parasitism behavior and combines with Lewy flight behavior in cuckoos [51], Gray Wolf Optimizer (GWO) is a powerful SI algorithm, which was inspired by the hunting mechanism and hierarchy mechanism of wolf [52]. The mathematical model is proposed based on the three leaderships including alpha, beta and delta. Moth-Flame Optimization (MFO) algorithm has been used for different fields such as marine propeller design, which is inspired by navigational phenomenon of moths in nature known as transverse orientation [53], Whale Optimization Algorithm (WOA) was proposed based on the social behavior of humpback whales, which involved three strategies including searching for prey, encircling prey and bubble-net attacking method [54].

Currently, many new SI algorithms have also been proposed. Aquila Optimizer (AO) is a novel SI algorithm inspired by hunting behaviors of Aquila, which use four methods for hunting [55]. Artificial Rabbits Optimization (ARO) is inspired by survival methods, where detour foraging and random hiding are used for exploration and exploitation, and energy shrink is utilized to switch from exploration and exploitation [56]. Snake Optimizer (SO) simulates the mating behavior of snake, and its exploration and exploitation depend on the food and environmental temperature [57]. Nutcracker Optimization Algorithm (NOA) was proposed according to two behaviors of Clark's nutcrackers, which store seeds in summer and autumn, and search for the previously stored seeds in winter and spring [58]. Shrimp and Goby Association (SGA) search algorithm inspired by the cooperation of shrimp and gobies was proposed, and applied for damage identification of Canton Tower in Guangzhou, China [59]. More SI algorithms are listed as Table 1.

Although SI algorithms are presented by different inspirations, exploration (diversification) and exploitation (intensification) are all involved in looking for the optimal solution in search space [72,73]. In the exploration phase, candidate solutions are utilized to globally explore the search space in order to look for the space of near-optimal solutions. The YUKI algorithm introduces a novel optimization

Table 1
SI algorithms presented in literature.

Algorithms	Inspiration	Year
Flower Pollination Algorithm (FPA) [60]	The pollination of flowers	2012
Bird Swarm Algorithm (BSA) [61]	Bird swarms	2016
Spotted Hyena Optimizer (SHO) [62]	Spotted hyenas	2017
Sunflower Optimization (SFO) [63]	Sunflowers	2018
Tree Growth Algorithm (TGA) [64]	Trees competition	2018
Squirrel Search algorithm (SSA) [65]	Southern flying squirrels	2019
Black Widow Optimization Algorithm (BWO) [66]	Black widow spiders	2020
Golden Eagle Optimizer (GEO) [67]	Golden eagles	2021
White Shark Optimizer (WSO) [68]	Great white sharks	2022
Dwarf Mongoose Optimization Algorithm (DMO) [69]	Dwarf mongoose	2022
Red piranha optimization (RPO) [70]	Red Piranha	2023
Fire Hawk Optimizer (FHO) [71]	Fire Hawks	2023

approach by employing a dynamic search space reduction technique [74]. This method entails modifying the search space size and placement in response to the search's progress, utilizing straightforward equations. Additionally, the algorithm incorporates two specialized systems, one for exploration and the other for exploitation behaviors, which work in tandem throughout the search [75–77]. In contrast, candidate solutions in the exploitation phase will search for the best solution in the potential region found in the exploration process. Parmaksiz et al. [78] modified Dragonfly algorithm (DA) by using mutation operation, boundary control, and greedy selection mechanisms to balance exploration and exploitation. Adhikary et al. [79] introduced three sequential reinforcement strategies combined with a social hierarchy mechanism and random walks using student's t-distributed random numbers to achieve a better balance between exploration and exploitation of gray Wolf Optimizer (GWO). Therefore, it is essential to balance the exploration and exploitation to find the best solution for optimization problems. A good strategy to balance these two phases is that algorithms should mainly emphasize on exploration in the early iterations and exploitation in the later iterations, but exploration and exploitation should be existed simultaneously in the entire iterations. However, achieving balance in these two phases is still a challenge due to the randomness of the search methods of SI algorithm [80]. Therefore, how to strike a better balance between exploration and exploitation is also the goal that researchers have been working on.

Many meta-heuristic algorithms including novel optimization algorithms [81,82], improved algorithms [83–85], and hybrid algorithms [86–88] have been emerged, and are applied for different fields such as feature selection [89,90], data clustering [91–93], image segmentation [94,95], neural network training [96,97], truss design [98], Unmanned Aerial Vehicle (UAV) path planning [99], and Traveling Salesman Problem (TSP) [100,101]. A survey called Supply chain challenges in 2021 from the Statista website shows that there are many challenges in the supply chain such as faster response time. In this regard, meta-heuristic algorithms can also be used to design and optimize supply chain networks [102], which indicates that they are very effective strategies for different designs. Also, as optimization problems increase and become more complex in the future, the demand for new optimization algorithms will also increase. In addition, No Free Lunch (NFL) theorem for optimization [103] shows that there are no optimization algorithms for solving all problems, so an optimization algorithm can be for a specific problem. In detail, even if one algorithm has a good optimization ability in some problems, its results in other problems may become unacceptable. Therefore, in this paper, a Sinh Cosh optimizer inspired by the characteristics of Sinh and Cosh functions in the interval of 0–1 is proposed to provide a new optimization strategy and better balance between exploration and exploitation. As far as we know, it is the first time that SCHO is proposed in the literature. The other contents of this paper are as follows.

Section 2 presents the inspiration and mathematical model of SCHO. Section 3 analyzes the results of the performance and practicality of SCHO for the 23 classical benchmark functions at different dimensions, CEC 2014 test functions and six engineering design problems compared to other well-known meta-heuristics. Section 4 shows the conclusion of SCHO and proposes the research direction for the future.

2. Sinh cosh optimizer (SCHO)

This section first presents the inspiration for SCHO. Afterward, the mathematical model of SCHO is introduced. It has four crucial components: the first and second phase of exploration and exploitation, the bounded search strategy, and the switching mechanism.

2.1. Inspiration

There are three points for inspiration for SCHO. Firstly, how to strike a balance between exploration and exploitation is still a big challenge, which indicates that more strategies need to be provided. Secondly, no algorithm can satisfy all optimization problems, as mentioned by NFL above. Therefore, new meta-heuristic algorithms still need to be proposed to face the complex and diverse problems. Finally, optimization algorithms based on mathematical inspiration, such as Sin Cosine Algorithm (SCA) [104] and Arithmetic Optimization Algorithm (AOA) [105], are proposed, which points out a new possible direction for researchers to study meta-heuristic algorithms. Hyperbolic functions are common types of trigonometric functions, among which sinh and cosh are the most basic hyperbolic functions. Two characteristics of Cosh and sinh can be used for meta-heuristic algorithms. One is that the values of cosh are always bigger than one, which is a critical boundary between exploration and exploitation. Another one is that the values of sinh are in the interval $[-1, 1]$ and close to zero, which can improve the exploration and exploitation.

According to these reasons, SCHO is proposed based on sinh and cosh, and its schematic diagram is shown in Fig. 1. The schematic diagram consists of three parts: the first phase of exploration and exploitation in Fig. 1(a), the second phase of exploration and exploitation in Fig. 1(b), and the bounded search strategy in Fig. 1(c). Yellow color represents the non-optimal space, which will be discarded in the first phase of search, while green represents the potential solution space. The potential candidates will be gathered to green space and carry out the second phase of deep search in it. The pink color represents the space of the bounded search strategy, and n represents the number of times to enter this strategy. When the iteration value is equal to the set value, the algorithm randomly distributes all candidate solutions in this space, and then executes the second phase of search strategy in this space. The cosh and sinh will be modified to achieve these strategies. The mathematical models will be introduced in the following subsections in details.

2.2. Mathematical model and algorithm

2.2.1. Initialization phase

Like other metaheuristic algorithms, SCHO also starts by randomly initializing a set of candidate solutions. The initialized candidate solutions are shown in Eq. (1), where the best candidate solution obtained during iterations is considered nearly the optimum.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,dim-1} & x_{1,dim} \\ x_{1,2} & \cdots & x_{2,j} & \cdots & x_{2,dim} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,dim} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,dim-1} & x_{N,dim} \end{bmatrix} \tag{1}$$

where X is a set of random candidate solutions obtained via using Eq. (2), $x_{i,j}$ denotes the j_{th} position of i_{th} solution, N represents the number of candidate solutions, and dim is the problem dimension.

$$X = rand(N, dim) \times (ub - lb) + lb \tag{2}$$

where $rand$ represents the random value in [0,1], ub and lb are the upper bound and lower bound of variables, respectively.

2.2.2. Exploration phase

SI algorithms emphatically balance the exploration and exploitation in the optimization process. Generally, these algorithms can explore the next position according to their position and the position of the best solution obtained so far. Therefore, in this study, exploring the next position is still related to its position, and the optimal solution obtained so far. Exploration is divided into two phases in the optimization process (iterations), and should exist in the later iterations to escape from local optima. The value to switch between these two phases is determined using Eq. (3):

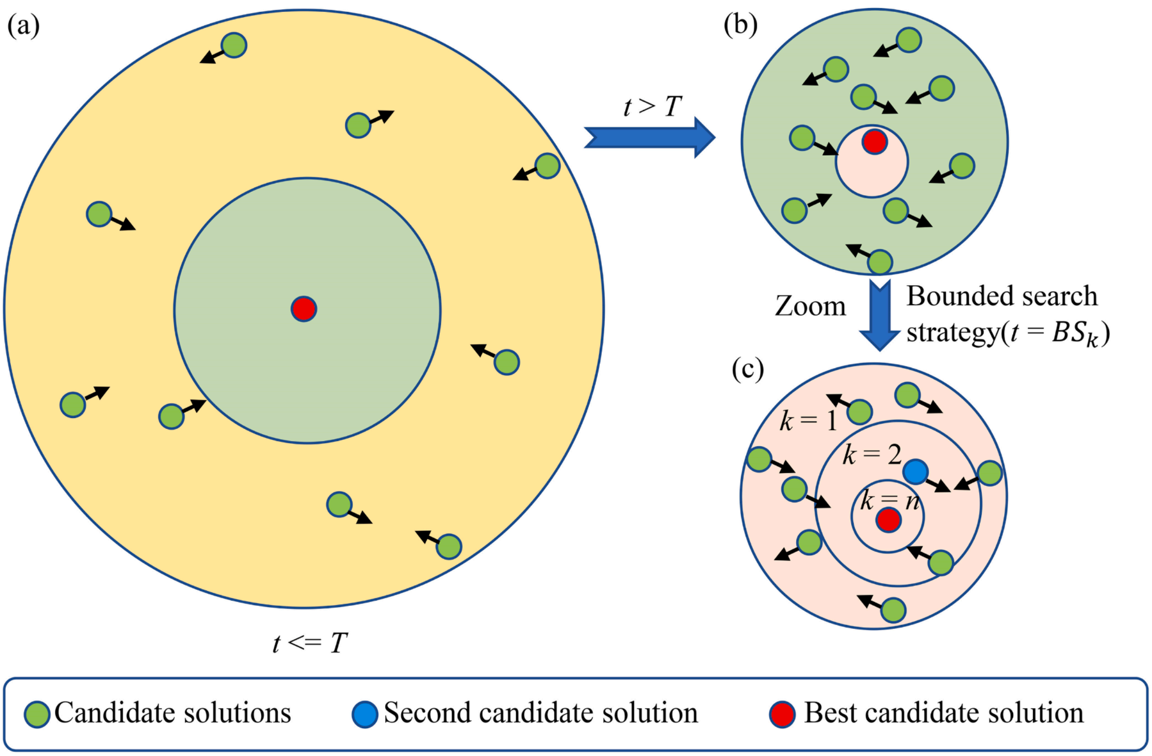


Fig. 1. SCHO schematic diagram: (a) the first phase of exploration and exploitation, (b) the second phase of exploration and exploitation and (c) the bounded search strategy.

$$T = floor\left(\frac{Max\_iteration}{ct}\right) \tag{3}$$

where $Max\_iteration$ denotes the maximum value of iterations, $floor$ is a function of rounding down in MATLAB, ct is a coefficient for setting switching point in two phases, fixed equal to 3.6 according to the experiments in this paper. In the first phase, the exterior of the search space near the position of search agents will be explored in the early iterations, and then search agents will gradually close to the best-obtained solution. The position update function for exploration in the first phase is proposed as in Eq. (4):

$$X_{(i,j)}^{t+1} = \begin{cases} X_{best}^{(j)} + r_1 \times W_1 \times X_{(i,j)}^{t}, & r_2 > 0.5 \\ X_{best}^{(j)} - r_1 \times W_1 \times X_{(i,j)}^{t}, & r_2 < 0.5 \end{cases} \tag{4}$$

where t indicates the current iteration, $X_{(i,j)}^{t}$ and $X_{(i,j)}^{t+1}$ mean the j_{th} position of the i_{th} solution in the current iteration and next iteration, respectively. $X_{best}^{(j)}$ is the j_{th} position of the optimal solution obtained as far, r_1/r_2 are the random numbers in the interval [0,1]. W_1 is the weight coefficient of $X_{(i,j)}^{t}$ in the first exploration phase, controlling candidate solutions in the first phase to stay away from themselves and gradually exploring towards the optimal solution, which is calculated using Eq. (5):

$$W_1 = r_3 \times a_1 \times (\cosh r_4 + u \times \sinh r_4 - 1) \tag{5}$$

where a_1 is a monotonically decreasing function, which is calculated by using Eq. (6). r_3/r_4 are random numbers in [0,1]. u is a sensitive coefficient that controls the accuracy of exploration in the first phase and is fixed equal to 0.388. Fig. 2 shows that W_1 gradually decreases from a small value, which indicates the importance of own position for position updates. Therefore, the candidate solutions gradually stay away from themselves in the first phase and then explore the optimal solution:

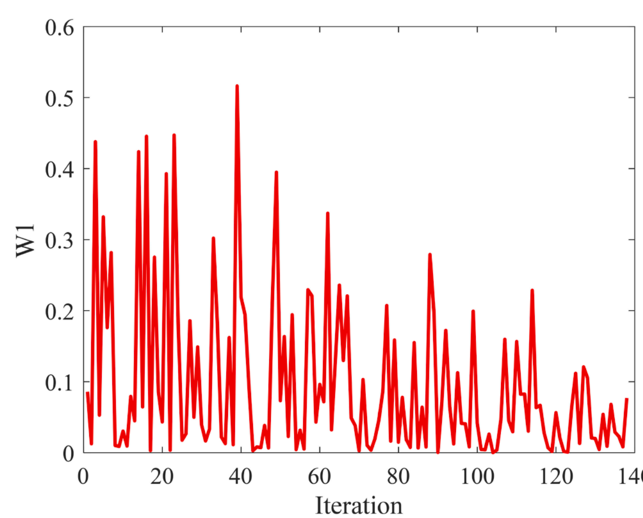


Fig. 2. The value of W_1.

$$a_1 = 3 \times \left(-1.3 \times \frac{t}{Max\_iteration} + m\right) \tag{6}$$

where m is the sensitive coefficient that controls the exploration accuracy of exploration and is equal to 0.45 based on the experiments in this paper.

In the second phase of exploration, the search agents are almost unaffected by the best-obtained solution and therefore they non-directionally explore the next position based on their current position. The position update function is calculated using Eq. (7):

$$X_{(i,j)}^{t+1} = \begin{cases} X_{(i,j)}^{t} + \left| \varepsilon \times W_2 \times X_{best}^{(j)} - X_{(i,j)}^{t} \right|, & r_5 > 0.5 \\ X_{(i,j)}^{t} - \left| \varepsilon \times W_2 \times X_{best}^{(j)} - X_{(i,j)}^{t} \right| & r_5 < 0.5 \end{cases} \tag{7}$$

where ε indicates a tiny positive number set to 0.003 based on the experiments in this paper. W_2 is the weight coefficient of $X_{best}^{(j)}$ in the second phase of exploration, which is calculated by using Eq. (8). Multiplying W_2 by ε greatly weakens the influence of the optimal solution on the current solutions, which causes undirected random exploration of candidate solutions around $X_{(i,j)}^{t}$:

$$W_2 = r_6 \times a_2 \tag{8}$$

where r_6 belongs to a random number in [0,1], a_2 indicates a monotonically decreasing function calculated by using Eq. (9), and W_2 is shown in Fig. 3:

$$a_2 = 2 \times \left(-\frac{t}{Max\_iteration} + n \right) \tag{9}$$

where n denotes a sensitive coefficient that controls the accuracy of exploration in the second phase of exploration and is set to 0.5 according to the experiments in this paper.

2.2.3. Exploitation phase

The exploitation of SI algorithms generally exists in the whole process (iterations). To fully exploit the search space, the exploitation is divided into two phases and takes place in the entire iterations. In the first exploitation phase, the near space of X is exploited, so the formula for exploitation is designed as Eq. (10):

$$X_{(i,j)}^{t+1} = \begin{cases} X_{best}^{(j)} + r_7 \times W_3 \times X_{(i,j)}^{t}, & r_8 > 0.5 \\ X_{best}^{(j)} - r_7 \times W_3 \times X_{(i,j)}^{t}, & r_8 < 0.5 \end{cases} \tag{10}$$

where r_7/r_8 belong to the random number in the interval [0,1]. W_3 is the weight coefficient for the first phase of exploitation and controls candidate solutions to exploit the search space around themselves from near to far, calculated using Eq. (11):

$$W_3 = r_9 \times a_1 \times (\cosh r_{10} + u \times \sinh r_{10}) \tag{11}$$

where r_9/r_{10} are random numbers in [0,1], a_1 is defined by Eq. (6), u is the same as the first exploration phase, fixed equal to 0.388. Fig. 4 shows W_3 gradually decrease from a larger value, which denotes that candidate solutions can exploit the space around themselves from near to far.

In the second phase of exploitation, candidate solutions will perform

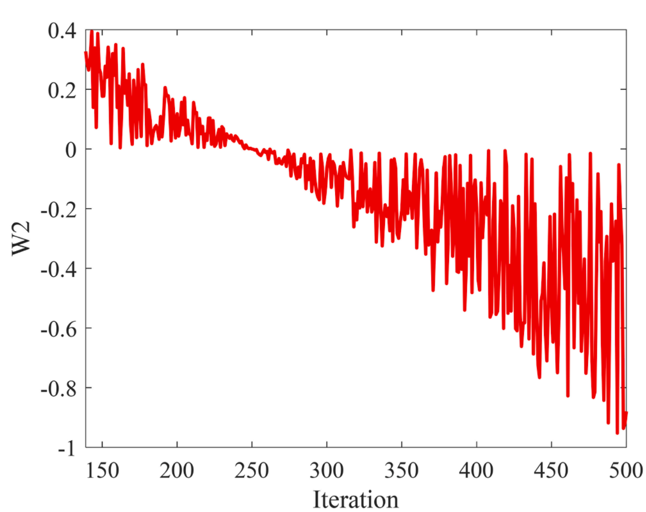


Fig. 3. The value of W_2.

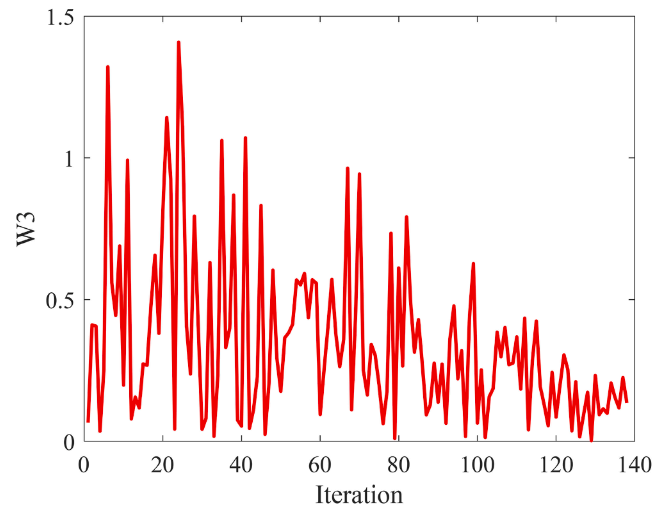


Fig. 4. The value of W_3.

deep exploitation around the optimal solution obtained as far, and the intensity of exploitation around the best-obtained solution will increase with increasing iterations. The position update function is shown in Eq. (12):

$$X_{(i,j)}^{t+1} = X_{(i,j)}^{t} + r_{11} \times \frac{\sinh r_{12}}{\cosh r_{12}} \left| W_2 \times X_{best}^{(j)} - X_{(i,j)}^{t} \right| \tag{12}$$

where r_{11}/r_{12} are random numbers in the interval [0,1], W_2 is calculated by using Eq. (8). W_2 controls the degree of the second exploitation phase. Its absolute value gradually increases in the later iterations, increasing the exploitation degree. The coefficient in front of the absolute value in Eq. (12) is used for keeping the diversity of candidate solutions.

2.2.4. Bounded search strategy

In order to fully exploit the potential search space, a strategy similar to animal hunting in the later stage is applied in SCHO, called the bounded search strategy. The potential search space is found by exploring the entire search area in the early iterations. To fully explore and exploit the potential space, all of candidate solutions are randomly initialized in this potential space by using Eq. (2). Then, the space will be deeply explored and exploited. Every start of this strategy is calculated by using Eq. (13):

$$BS_{k+1} = BS_k + floor\left(\frac{Max\_iteration - BS_k}{\alpha} \right) \tag{13}$$

where k is a positive integer starting from 1, BS_k is calculated using Eq. (14) and BS_{k+1} denotes the numbers of iterations that start the current and next bounded search strategy, respectively. α denotes a sensitive coefficient and controls the accuracy of deep exploration and exploitation in the potential space, which is set to 4.6 based on the experiments in this paper.

$$BS_1 = floor\left(\frac{Max\_iteration}{\beta} \right) \tag{14}$$

where β controls the value that starts the bounded search strategy and is set to 1.55.

When SCHO uses the bounded search strategy every time, the upper and lower bound of optimization problems will be calculated using Eq. (15) and Eq. (16), respectively:

$$ub_k = X_{best}^{(j)} + \left(1 - \frac{t}{Max\_iteration} \right) \times \left| X_{best}^{(j)} - X_{second}^{(j)} \right| \tag{15}$$

$$lb_k = X_{best}^{(j)} - \left(1 - \frac{t}{Max\_iteration}\right) \times \left|X_{best}^{(j)} - X_{second}^{(j)}\right| \tag{16}$$

where ub_k and lb_k represent the upper and lower bound of the potential search space, $X_{second}^{(j)}$ indicates the j_{th} position of the suboptimal solution.

2.2.5. Switching mechanism

In the SCHO, a switching mechanism with Sinh and Cosh is proposed to switch between the exploration and exploitation. To achieve the exploration and exploitation of the entire search space and escape from the local optimum in the later iteration, the switching mechanism should mainly focus on the exploration, but performs a small amount of exploitation in the early iterations. In contrast, in the later iterations, the switching mechanism should mainly focus on the exploitation but performs a small amount of exploration. The principle used to design the switching mechanism is shown in Eq. (17):

$$A = \left(p - q \times \left(\frac{t}{Max\_iteration}\right)^{\left(\frac{\cosh\frac{t}{Max\_iteration}}{\sinh\frac{t}{Max\_iteration}}\right)}\right) \times r_{13} \tag{17}$$

where r_{13} belongs to a random number in the interval [0,1], and p and q are the balance coefficient for controlling the exploration and exploitation during iterations, which are equal to 10 and 9, respectively. When $A > 1$, SCHO performs the exploration, while when $A < 1$, SCHO performs the exploitation. The values of A are shown in Fig. 5.

Fig. 5 shows that SCHO mainly performs the exploration in the early iterations. With increasing iterations, SCHO gradually focuses on the exploitation, but keeps the exploration in the later iteration, allowing SCHO to escape from the local optimum in the later iterations.

To recap, SCHO is divided into four parts, including the first phase ($t < T$) and second phase ($t > T$) of exploration and exploitation, bounded search strategy ($t = BS_{k+1}$), and the switching mechanism (A). Firstly, SCHO starts optimization via randomly initializing a set of candidate solutions, and then explores the search area from search agents toward the optimal solution obtained as far or exploits the search area around the search agents from near to far when $t < T$. Otherwise, SCHO enters the second phase and performs deep exploration and exploitation of the potential search area. Also, when $t = BS_{k+1}$, SCHO uses the bounded search strategy to distribute all search agents into the potential space again, and continues performing the second phase of exploration and exploitation. Finally, the switching mechanism is utilized to switch the exploration ($A > 1$) and exploitation ($A < 1$). The pseudo-code of the proposed SCHO is detailed in Algorithm 1. The detailed flowchart of SCHO is presented in Fig. 6.

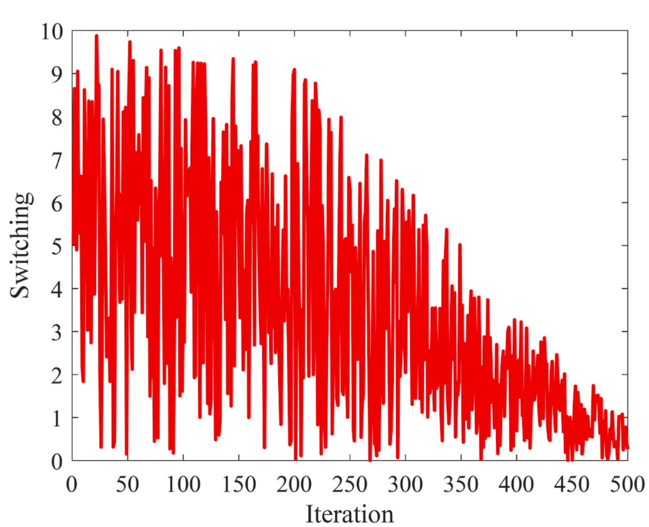


Fig. 5. The values of A, the switching mechanism parameter, with 500 iterations.

2.3. Computational complexity of SCHO

The computational complexity of SCHO is related to initialization process, fitness function evaluation, bounded search strategy and the updating position of solutions. The complexity of initialization process is of $O(N)$, where N shows the number of candidate solutions. The complexity of fitness function evaluation is of $O(T \times N \times c)$, where T denotes iterations and c is the function evaluation's cost. The complexity of bounded search strategy includes two parts: the bubble sort ($O(z \times N^2)$) and the redistribution process of candidate solution ($O(z \times N^2 \times D)$), where z indicates the number of times that SCHO enters the process of bounded search strategy and D represents the dimension of this problem. Finally, the complexity of the updating position of candidate solutions is of $O(T \times N \times D)$. Therefore, the total computational complexity of SCHO can be given as:

$$\begin{aligned} O(SCHO) &= O(N) + O(T \times N \times c) + O\left(z \times N^2\right) + O\left(z \times N^2 \times D\right) \\ &\quad + O(T \times N \times D) \\ &= O\left(N + T \times N \times c + T \times N \times D + z \times N^2 + z \times N^2 \times D\right) \end{aligned} \tag{18}$$

Since $N \ll T \times N \times D$, $z \times N^2 \ll z \times N^2 \times D$, the total computational complexity can be reduced to Eq. (19)

$$O(SCHO) \cong O\left(T \times N \times c + T \times N \times D + z \times N^2 \times D\right) \tag{19}$$

Algorithm 1
Pseudo-code of the Sinh Cosh Optimizer (SCHO).

```
Initialize SCHO parameters ct, T, BS_1, u, m, n, α, β, p, q.
Initialize the positions of candidate solutions randomly. X: i = 1, …, N.
While (t < Max_iteration) do
  Calculate the Fitness values of the candidate solutions (X).
  Find the Optimal solution so far.
  for (i = 1 to N) do
  for (j = 1 to dim) do
    Update A by using Eq. (17).
    if (t = BS_k) then
      Find the position of the second solution currently.
      Update BS_k by using Eq. (13).
      Update the search space by using Eq. (15) and Eq. (16).
      Distribute all candidate solutions by using Eq. (2).
   end
   if (A > 1) then
      Enter the exploration phase
      Update W_1 and W_2 by using Eq. (5) and Eq. (8), respectively.
      if (t <= T) then
      The first phase of exploration.
      Update the position of candidate solutions by using Eq. (4).
      else
      The second phase of exploration.
      Update the position of candidate solutions by using Eq. (7)
   else
   Enter the exploitation phase
   Update W_3 value by using Eq. (11).
   if (t <= T) then
      The first phase of exploitation.
      Update the position of candidate solutions by using Eq. (10).
   else
      The second phase of exploitation.
      Update the position of candidate solutions by using Eq. (12).
   end
   end
   t = t + 1
end while
Return the best solution (B(X)).
```

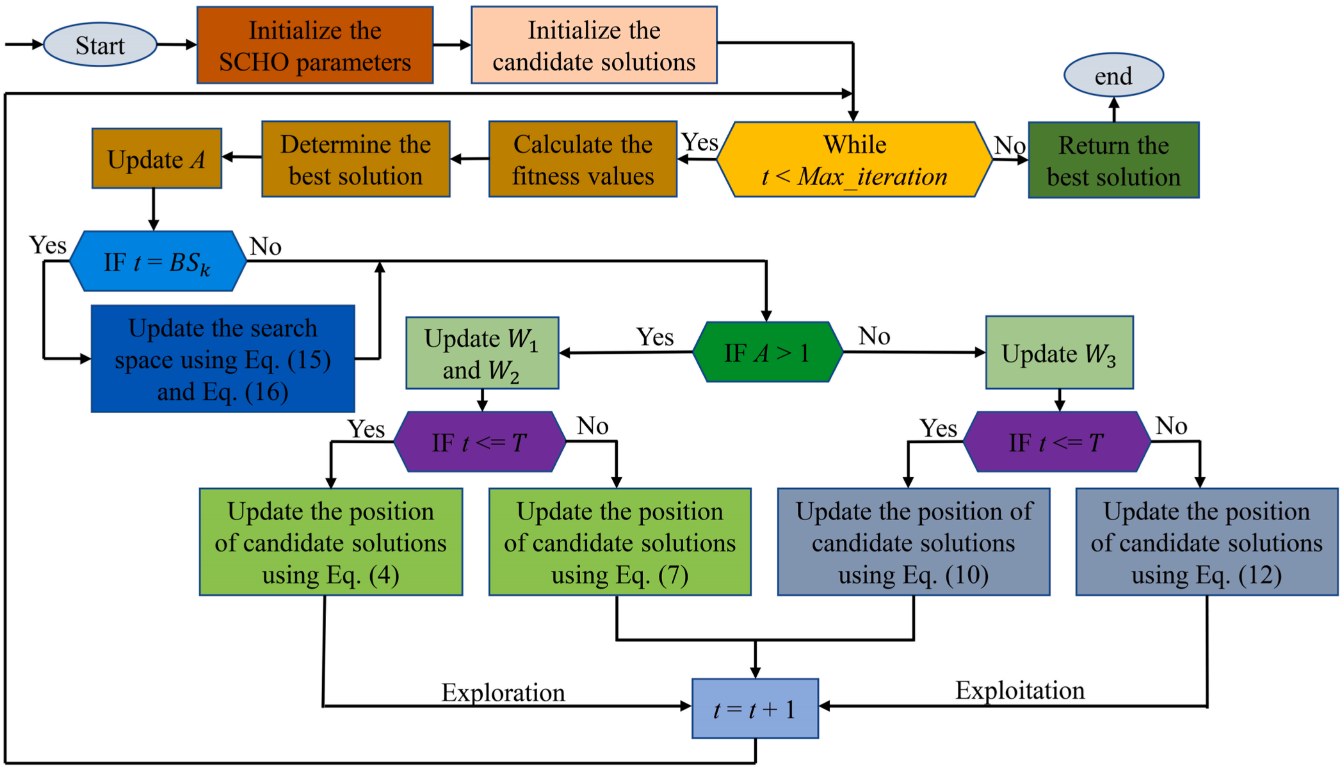


Fig. 6. Detailed flowchart of SCHO.

3. Results & discussions

In this section, the robustness and efficiency of SCHO are tested for the 23 classical benchmark functions, CEC 2014 test functions and six engineering design problems, respectively.

3.1. Optimization results of SCHO on benchmark test functions

Twenty-three classical benchmark functions are generally applied to test the performance of meta-heuristic algorithms. They are divided into three families: unimodal, multi-modal, and composite test functions. The first family shown in Table 2 has only one global optimal solution and is utilized to evaluate optimization algorithms' convergence and exploitation ability. The second family shown in Table 3 has many local optima except one optimal global solution and is used to test the ability of local optimum avoidance and the exploration ability of algorithms. The final group shown in Table 4 includes ten fixed-dimensional multi-modal functions.

3.1.1. Analysis of four subordinate models of SCHO

To study the effect of the exploration (the first subordinate model), the exploitation (the second subordinate model), the boundary search strategy (the third subordinate model), and the switching mechanism (the fourth subordinate model) on the optimization performance of SCHO, SCHO and its five variants are executed for solving 23 classical

Table 2
Unimodal test functions.

Function	Dimension	Range	f_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=0}^n \lvert x_i\rvert + \prod_{i=0}^n \lvert x_i\rvert$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^d (\sum_{j=1}^i x_j)^2$	30	[−100, 100]	0
$F_4(x) = max\{\lvert x_i\rvert,\ 1 \le i \le n\}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1}[100(x_i^2 - x_{i-1})^2 + (1 - x_i)^2]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[−100, 100]	0
$F_7(x) = \sum_{i=0}^n ix_i^4 + random[0,\ 1)$	30	[−1.28, 1.28]	0

Table 3
Multi-modal test functions.

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

Table 4
Fixed-dimension multi-modal test functions.

Function	Dimension	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	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
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	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
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−5, 5]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i\ exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[0, 1]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363

benchmark functions. Five variants are SCHO without the third subordinate model (SCHO_NT), SCHO without the second, third and fourth subordinate models (SCHO_NSTF), SCHO without the first, third and fourth subordinate models (SCHO_NFTF), SCHO without the second and fourth subordinate models (SCHO_NSF), and SCHO without the first and fourth subordinate models (SCHO_NFF). For these tests, the number of search agents (N) and iterations (*Max_interation*) are specified as 30 and 500, respectively, and then these algorithms run for 15,000 fitness evaluations to find the optimal solution. The dimension (D) of these benchmark functions (F1-F13) is specified as 30. Each algorithm on each tested function is run independently 30 times to ensure that optimization results are not just a random occurrence.

Table 5 shows the optimization results of SCHO, SCHO_NT, SCHO_NSTF, SCHO_NFTF, SCHO_NSF, and SCHO_NFF. The Friedman ranking test is executed to rank the average fitness obtained by different SCHO in a statistical method. The Friedman ranking results show that SCHO_NFTF exhibits the worst optimization ability, because SCHO with only exploitation is prone to being trapped in local optima. It can be observed that the optimization ability of SCHO_NSTF is only better than that of SCHO_NFTF among the six algorithms. The results of SCHO_NFF and SCHO_NSF indicate that the performance of SCHO_NFTF and SCHO_NSTF is improved using the boundary search strategy. It can also be seen that SCHO, SCHO_NFF and SCHO_NSF with the boundary search strategy can obviously enhance the optimization performance for F6, F8, F13, and F21-F23 compared with SCHO_NT, SCHO_NSTF, and SCHO_NFTF. The result of SCHO_NT shows that the performance of algorithm combining exploration and exploitation outperforms the algorithm with only exploration or exploitation. In addition, SCHO wins all changed algorithms with different subordinate models, which shows that the synergy of four subordinate models can achieve the best optimization performance.

To show the importance of sinh & cosh in these subordinate models, Eq. (12) and Eq. (17) were selected as examples to verify them for 23 benchmark functions. SCHO is compared with two variants of SCHO to verify the importance of the equation in the second exploitation phase. The first one is to use the first phase of exploitation equation in two exploitation phase (variant 1), and another one is the variant without the second phase of exploitation equation (variant 2). The results are shown in Fig. 7(a). It can be seen that SCHO wins two variants in terms of the number of the average optimal solutions, which shows the importance of Eq. (12) for the performance of SCHO. To verify the effect of sinh & cosh on the equations in the subordinate models of exploration and exploitation, the optimization results of SCHO are compared with that of four variants. The first two variants are to use the sine & cosine with the same value range to replace the sinh & cosh in second phase of exploitation equation (variant 3) and the sinh & cosh in the first phase of exploration equation (variant 4). The latter two variants are to use the linear functions with the same value range to replace the sinh & cosh in second phase of exploitation equation (variant 5) and the sinh & cosh in the first phase of exploration equation (variant 6). The results are shown in Fig. 7(b). It can be observed that SCHO outperforms these variants with different non-hyperbolic functions in terms of the number of the average optimal solutions, and sinh & cosh have a greater impact in exploration than in exploitation. To further evaluate the effect of the switching mechanism on the performance of SCHO, three variants with different switch equation are used, and the optimization results of SCHO are compared with that of them. These variants are the switching mechanism using sine & cosine (variant 7), the switching mechanism using linear functions (variant 8), and the switching mechanism with same selection probability for exploration and exploitation (variant 9). Fig. 7(c) shows that SCHO wins three variants, which indicates that the switching mechanism with sinh & cosh is more beneficial for the performance of SCHO. Similarly, SCHO are compared with the variant with sine & cosine in Eq. (17) (variant 10) and the variant with linear functions in Eq. (17) (variant 11) to verify the impact of sinh & cosh in Eq. (17) for the performance of SCHO. Fig. 7(d) shows that SCHO is superior to other variants, which illustrates that sinh & cosh is more suitable for the switching mechanism of SCHO. In short, these subordinate models are crucial to the performance of SCHO, and combining sinh & cosh with these subordinate models can more effectively improve the performance of SCHO.

3.1.2. Qualitative analysis of SCHO

In order to validate the performance of SCHO, the convergence of SCHO is analyzed by utilizing trajectories and convergence curves, as shown in Fig. 8. There are five qualitative metrics for the convergence of SCHO, containing the 2D graph of the functions mentioned above (as in the first column) to observe the topology of these functions, the search history of search agents (as in the second column), the trajectories of the first dimension of the first agent (as in the third column), the average fitness value of all search agents (as in the fourth column) and the convergence curve of the optimal solution obtained in each iteration (as in the last column).

In general, search history is performed to study the collective behavior of search agents and the interaction between search agents and optimal solution, which provides a useful method to observe the modality of swarm search of search agents. The modality can present a search process in which search agents always surround the best solution for unimodal functions. In contrast, it can present the scattering of search agents on multi-modal functions. These modality properties can be utilized to improve the performance of exploration and exploitation when SCHO is applied for solving the unimodal and multi-modal functions.

The trajectories shown in Fig. 8 indicate that the search agent has a high frequency and amplitude in the early iteration, but nearly vanish soon, which denotes that SCHO possesses the strong abilities of exploration in the early iterations and exploitation in the later iterations. Moreover, it can also be observed from Fig. 8 that the average fitness value of search agents is very high at the first few iterations. Still, the average fitness gets very low soon, indicating that SCHO can converge to the optimal solution with a small number of iterations. In addition, the

Table 5
Optimization results of SCHO and SCHO without partial subordinate models for the 23 classical benchmark functions.

Fun	Measure	SCHO	SCHO_NT	SCHO_NSTF	SCHO_NFTF	SCHO_NSF	SCHO_NFF
F1	Best	0.000E+00	0.000E+00	0.000E+00	2.659E-14	0.000E+00	1.158E-16
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.570E-12	**0.000E+00**	2.672E-12
	STD	0.000E+00	0.000E+00	0.000E+00	5.316E-12	0.000E+00	6.116E-12
	Rank	1	1	1	6	1	5
F2	Best	0.000E+00	0.000E+00	0.000E+00	4.824E-09	0.000E+00	4.384E-10
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	1.437E-08	**0.000E+00**	8.697E-09
	STD	0.000E+00	0.000E+00	0.000E+00	8.286E-09	0.000E+00	9.703E-09
	Rank	1	1	1	6	1	5
F3	Best	0.000E+00	0.000E+00	0.000E+00	4.276E-05	0.000E+00	1.340E-06
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	2.029E-01	**0.000E+00**	1.390E-01
	STD	0.000E+00	0.000E+00	0.000E+00	3.423E-01	0.000E+00	4.302E-01
	Rank	1	1	1	6	1	5
F4	Best	0.000E+00	0.000E+00	0.000E+00	1.616E-04	0.000E+00	2.048E-07
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.637E-03	**0.000E+00**	7.026E-04
	STD	0.000E+00	0.000E+00	0.000E+00	4.773E-03	0.000E+00	9.504E-04
	Rank	1	1	1	6	1	5
F5	Best	2.156E+01	2.529E+01	2.647E+01	2.641E+01	2.661E+01	2.643E+01
	Average	2.818E+01	2.811E+01	2.874E+01	2.783E+01	2.862E+01	**2.781E+01**
	STD	1.388E+00	7.644E-01	5.677E-01	7.265E-01	6.209E-01	7.571E-01
	Rank	4	3	6	2	5	1
F6	Best	6.853E-06	1.418E+00	1.775E+00	1.008E+00	5.001E-04	2.606E-06
	Average	6.570E-01	2.576E+00	3.286E+00	2.393E+00	1.232E+00	**6.259E-01**
	STD	1.205E+00	6.778E-01	7.275E-01	4.878E-01	1.811E+00	1.188E+00
	Rank	2	5	6	4	3	1
F7	Best	3.031E-07	1.646E-06	2.766E-06	1.079E-03	5.818E-06	2.300E-06
	Average	**7.236E-05**	8.562E-05	7.886E-05	7.095E-03	7.714E-05	2.622E-03
	STD	8.953E-05	8.871E-05	9.120E-05	4.139E-03	6.096E-05	3.762E-03
	Rank	1	4	3	6	2	5
F8	Best	−1.257E+04	−6.947E+03	−6.678E+03	−5.518E+03	−1.255E+04	−1.255E+04
	Average	**7.661E+03**	−5.572E+03	−5.877E+03	−4.121E+03	−7.526E+03	−6.490E+03
	STD	2.540E+03	6.336E+02	5.987E+02	4.693E+02	2.587E+03	3.440E+03
	Rank	1	5	4	6	2	3
F9	Best	0.000E+00	0.000E+00	0.000E+00	8.527E-13	0.000E+00	0.000E+00
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.747E+00	**0.000E+00**	1.073E+00
	STD	0.000E+00	0.000E+00	0.000E+00	9.106E+00	0.000E+00	3.762E+00
	Rank	1	1	1	6	1	5
F10	Best	4.441E-16	4.441E-16	4.441E-16	3.141E-08	4.441E-16	1.438E-09
	Average	**4.441E-16**	**4.441E-16**	**4.441E-16**	2.817E-07	**4.441E-16**	1.453E-07
	STD	0.000E+00	0.000E+00	0.000E+00	1.910E-07	0.000E+00	1.493E-07
	Rank	1	1	1	6	1	5
F11	Best	0.000E+00	0.000E+00	0.000E+00	4.674E-14	0.000E+00	0.000E+00
	Average	**0.000E+00**	**0.000E+00**	**0.000E+00**	1.164E-02	**0.000E+00**	2.323E-03
	STD	0.000E+00	0.000E+00	0.000E+00	1.459E-02	0.000E+00	5.365E-03
	Rank	1	1	1	6	1	5
F12	Best	9.755E-10	6.905E-02	3.762E-01	1.237E-01	1.586E-06	3.125E-11
	Average	1.146E-01	3.415E-01	1.017E+00	1.972E-01	5.620E-01	**6.625E-07**
	STD	2.405E-01	2.719E-01	3.426E-01	4.574E-02	6.432E-01	6.949E-07
	Rank	2	4	6	3	5	1
F13	Best	7.953E-09	1.331E+00	1.470E+00	1.452E+00	2.850E-05	1.592E-06
	Average	1.266E+00	2.030E+00	2.537E+00	1.802E+00	1.801E+00	**1.044E+00**
	STD	9.275E-01	3.328E-01	4.190E-01	2.154E-01	1.243E+00	8.715E-01
	Rank	2	5	6	4	3	1
F14	Best	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01
	Average	3.8285	6.61	6.1164	3.4879	5.0945	**2.5037**
	STD	4.577E+00	4.418E+00	4.517E+00	3.234E+00	5.139E+00	2.923E+00
	Rank	2	6	5	3	4	1
F15	Best	3.082E-04	3.077E-04	3.075E-04	3.101E-04	3.075E-04	3.117E-04
	Average	**3.254E-04**	7.537E-04	6.490E-03	4.932E-04	4.425E-03	4.063E-04
	STD	2.572E-05	2.697E-04	8.791E-03	1.589E-04	8.107E-03	9.263E-05
	Rank	1	4	6	3	5	2
F16	Best	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00
	Average	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	STD	2.443E-08	2.449E-08	1.601E-08	1.214E-06	1.594E-08	1.419E-06
	Rank	1	1	1	1	1	1
F17	Best	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	Average	0.39789	0.39789	0.39789	0.39811	0.39789	0.39807
	STD	2.429E-06	2.938E-06	1.985E-06	2.592E-04	2.702E-06	2.455E-04
	Rank	1	1	1	6	1	5
F18	Best	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00
	Average	3.5303	3.0735	7.4332	**3**	5.8866	**3**
	STD	2.157E+00	3.712E-01	1.630E+01	1.138E-05	6.496E+00	1.311E-05
	Rank	4	3	6	1	5	1
F19	Best	−3.863E+00	−3.863E+00	−3.863E+00	−3.863E+00	−3.863E+00	−3.863E+00
	Average	**−3.8625**	**−3.8625**	**−3.8625**	−3.8601	**−3.8625**	−3.8603

(continued on next page)

Table 5 (*continued*)

Fun	Measure	SCHO	SCHO_NT	SCHO_NSTF	SCHO_NFTF	SCHO_NSF	SCHO_NFF
	STD	1.433E-03	1.437E-03	1.422E-03	3.381E-03	1.438E-03	3.210E-03
	Rank	1	1	1	6	1	5
F20	Best	−3.322E+00	−3.322E+00	−3.322E+00	−3.320E+00	−3.322E+00	−3.319E+00
	Average	−3.2652	−3.2653	−3.2694	−3.2111	**−3.2728**	−3.1925
	STD	7.399E-02	1.097E-01	6.669E-02	9.971E-02	6.116E-02	1.396E-01
	Rank	4	3	2	5	1	6
F21	Best	−1.015E+01	−1.015E+01	−1.015E+01	−9.925E+00	−1.015E+01	−1.015E+01
	Average	−9.3025	−6.1315	−6.1228	−6.8198	**−9.4741**	−8.1314
	STD	1.921E+00	3.610E+00	3.253E+00	2.670E+00	1.747E+00	2.751E+00
	Rank	2	5	6	4	1	3
F22	Best	−1.040E+01	−1.040E+01	−1.040E+01	−9.957E+00	−1.040E+01	−1.040E+01
	Average	−9.7879	−5.9505	−6.4713	−7.4152	**−9.8646**	−8.8038
	STD	1.889E+00	3.493E+00	3.569E+00	2.661E+00	1.615E+00	2.732E+00
	Rank	2	6	5	4	1	3
F23	Best	−1.054E+01	−1.053E+01	−1.052E+01	−1.035E+01	−1.054E+01	−1.054E+01
	Average	−9.9194	−5.8951	−5.8011	−6.9884	−9.3805	−8.6732
	STD	1.902E+00	3.655E+00	3.719E+00	2.779E+00	2.372E+00	2.913E+00
	Rank	1	5	6	4	2	3
Mean	Rank	1.65	2.96	3.35	4.52	2.13	3.35
Final	Ranking	1	3	4	6	2	4

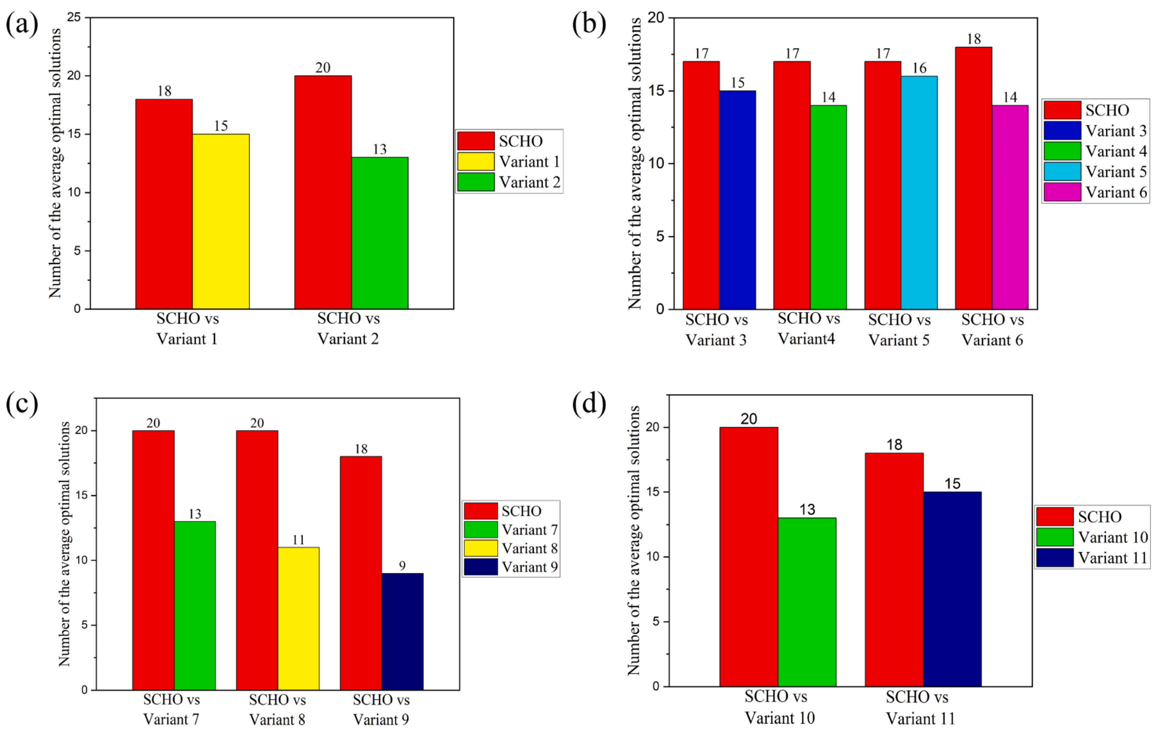


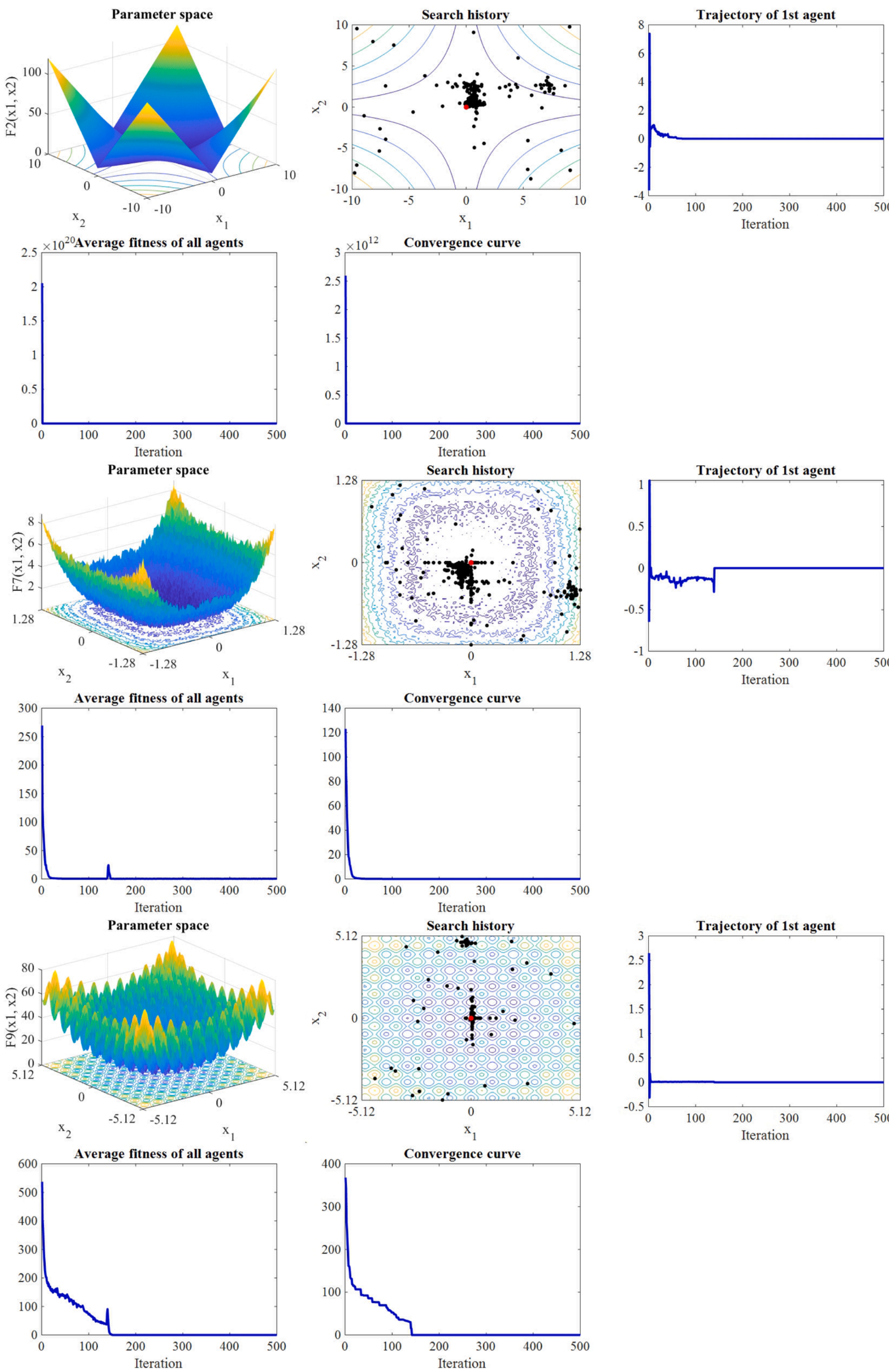
Fig. 7. (a) Comparison results between SCHO and two variants to verify the impact of the second exploitation phase for SCHO. (b) Comparison results between SCHO and variants with sine & cosine or linear function parameters in Eq. (12) to verify the impact of sinh & cosh for exploration and exploitation. (c) Comparison results between SCHO and variants with different switching mechanism to verify the significance of the proposed switching mechanism. (d) Comparison results between SCHO and variants with sine & cosine or linear function parameters in Eq. (17) to verify the impact of sinh & cosh for SCHO.

average fitness value has a transient increase in most functions when the iteration reaches about 140, mainly caused by the switch from the first phase to the second phase of exploration and exploitation. Therefore, when SCHO enters the second phase, its exploration ability has a temporary enhancement to escape from the local optima, which can be verified from the average fitness value that becomes smaller for some functions.

Finally, it is noted that the convergence of SCHO in the last column in Fig. 8 has a different pattern on these functions mentioned above. The convergence curve is smooth on the unimodal tested functions, but on the multi-modal tested functions, is more like a stepwise behavior because multi-modal tested functions are more intricate than unimodal tested functions. Furthermore, due to the second phase of exploration and exploitation and the bounded search strategy in the later iterations, the fitness values obtained by SCHO are closer to optimal solutions for these test functions. For example, F9 and F10 clearly obtain better solutions when entering the second phase of exploration and exploitation, while F21 also obtains a smaller solution due to the bounded search strategy, which indicates that both strategies can effectively help the search solution to escape from the local optimum. In short, SCHO can make a balance between exploration and exploitation well enough, and has a high chance of getting close to the optimal solution.

3.1.3. Comparison of optimization results between SCHO and other algorithms

To assess the performance of SCHO for the 23 classical benchmark



Continued

Fig. 8. Qualitative results of SCHO.

functions, the results obtained by SCHO are compared with six well-know and two new meta-heuristic optimization algorithms, including GWO [52], ALO [106], SCA [104], SSA [107], AOA [105], RSA [108], SHO [109], GJO [80]. The popular or new algorithms with same benchmark functions for evaluation are chosen to guarantee the reliability and superiority of SCHO. The important parameters of comparison algorithms are all specified, as shown in Table 6. For these tests, the number of search agents (*N*), iterations (*Max_interation*) and the dimension (*D*) are set to the same values as before. Each algorithm on each tested function is run independently 30 times to ensure that optimization results are not just a random occurrence.

Moreover, several metrics are applied for evaluating the performance of SCHO on these tested functions, including the best fitness value, average fitness value, and standard deviation (STD). Also, the

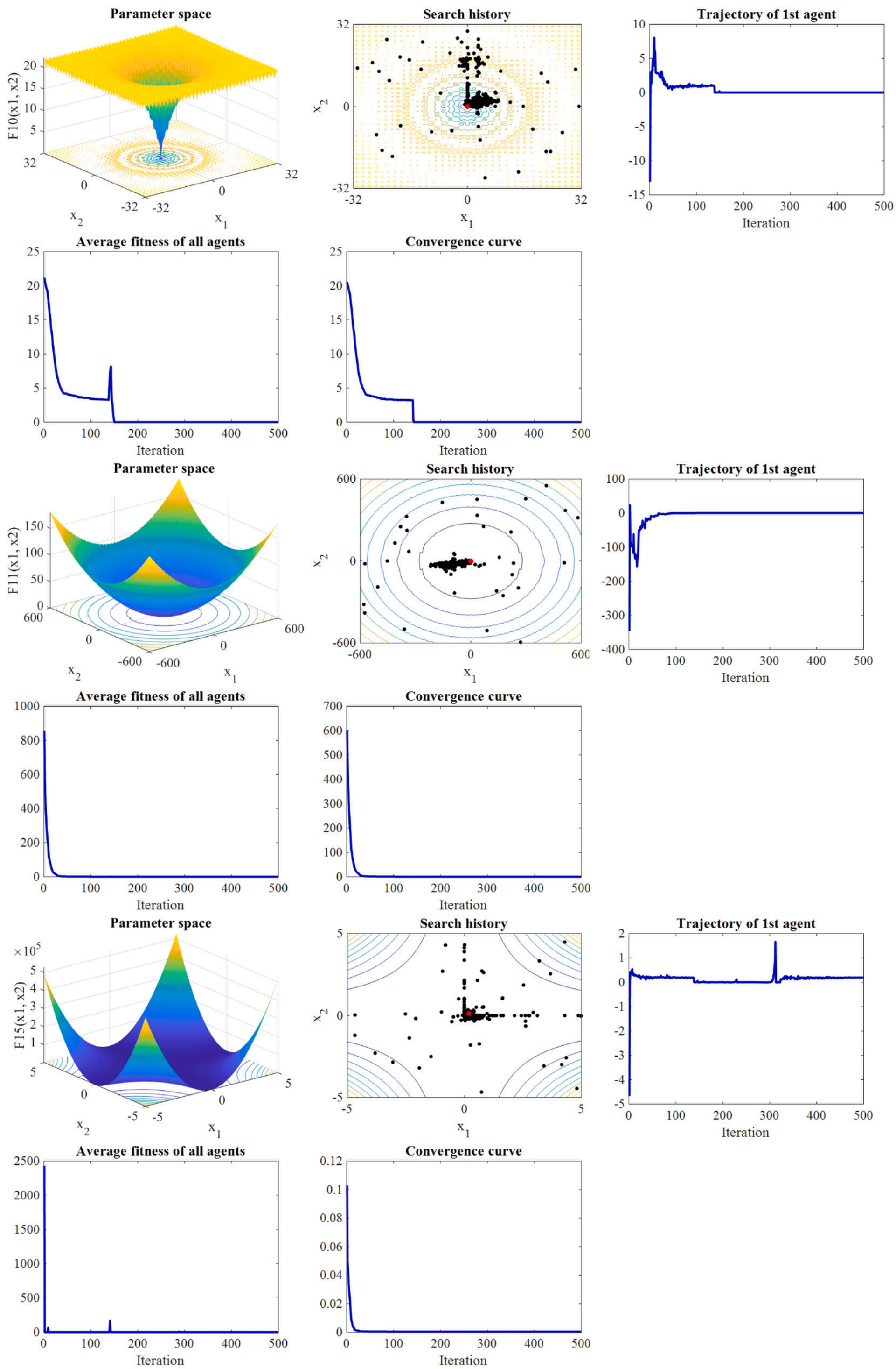


Fig. 8. (*continued*).

Friedman ranking test is implemented to rank the average fitness obtained by algorithms in a statistical method, and the mean ranking of SCHO and comparison algorithms is finally calculated to assess the performance of SCHO on each tested function. These results obtained by SCHO and other comparison algorithms are shown in Table 7.

The results of the Friedman ranking test indicate that SCHO is ranked first in comparison to other algorithms on the 23 classical benchmark functions, followed by GJO, which is ranked second, and GWO, SHO, RSA, SSA, ALO, AOA, SCA, which are respectively ranked third, fourth, fifth, sixth, seventh, eighth, and ninth. It is also seen that SCHO obtained the average optimal solution on fourteen tested functions (F1-F4, F7-F11, F15-F17, F19, F21), which account for 61%, ranked first, followed by RSA, which is ranked second with nine average optimal solutions. Then SSA, GJO, GWO, ALO, SHO, AOA, and SCA obtained the six, five, five, five, four, four, and two optimal solutions, respectively. Also, SCHO still obtain the six best solutions (F12, F14, F18, F20, F22, F23) out of the nine test functions that SCHO did not obtain the average optimal solution. In addition, SCHO and RSA are ranked first with five average optimal solutions on the unimodal functions (F1-F7). Still, the mean ranking of SCHO is better than RSA, which indicates that SCHO

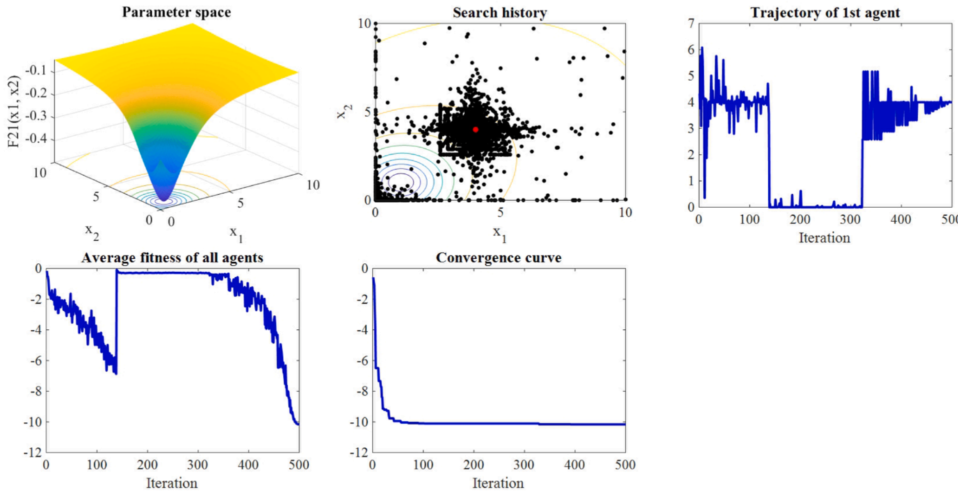


Fig. 8. (*continued*).

Table 6
Parameters of SCHO for different algorithms.

Algorithm	Parameter	Value
SCHO	*ct, u, m*	3.6, 0.388, 0.45
	ϵ, *n*	0.003, 0.5
	α, β	4.6, 1.55
	p, *q*	10, 9
GWO	*a* (convergence parameter)	Linear reduction from 2 to 0
ALO	*w* (integer)	[2,6]
SCA	*a* (*constant*)	2
SSA	c2, c3 (random numbers)	[0, 1]
AOA	α (sensitive parameter)	5
	u (control parameter)	0.5
RSA	α (sensitive parameter)	0.1
	β (sensitive parameter)	0.005
SHO	r_1 (*the cut − off point*)	0
	probability of success r2	0.1
GJO	c_1 (*constant*)	1.5
	β (*constant*)	1.5

presents the best exploitation ability compared to other comparative algorithms. Moreover, on the multi-modal tested functions (F8-F13), SCHO and RSA found four average optimal solutions and are ranked first. Still, SCHO is better in terms of the mean ranking than RSA, which shows that SCHO has a better exploration than other comparative algorithms. According to NFL, algorithms are specific to problems. On the 23 classical benchmark functions, SCHO provides a very nice method for solving F15 compared to these mentioned algorithms and other new algorithms such as Chimp Optimization Algorithm (ChOA) [5] and Goby Association Search algorithm (SGA) [59], which indicates SCHO can strike well balance between exploration and exploitation for F15. Therefore, SCHO provides a good balance strategy to get the optimal solution compared to other algorithms, which verify the effectiveness of proposed switching mechanism.

To evaluate the significant difference between SCHO and other algorithms, another statistical method called Wilcoxon rank-sum test is performed [6,110]. A significant level of α is set to 0.05, and the performance of SCHO and other algorithms is tested using the p value. Table 8 shows the results of Wilcoxon rank sum test between SCHO and the other algorithms. When $p > \alpha$, there is no significant difference between SCHO and the other algorithms, and the result is marked by '$\sim$'. When $p < \alpha$, there is a significant difference between SCHO and the other algorithms, and '+' denotes that SCHO gets better results than other algorithms and '-' declares SCHO gets a worse result compared to other algorithms. It can be seen from the table 8 that SCHO still get better results compared with other algorithms, which further shows the reliability and effectiveness of SCHO.

To observe the entire optimization process between all of algorithms more intuitively, the convergence curves of SCHO and other algorithms for the 23 tested functions are shown in Fig. 9. It is observed that SCHO gets the optimal solution on most tested functions (F1-F4, F9-F11, F16, F17, F19) with a fast convergence speed, especially on functions F16 and F17. Although SCHO obtains the optimal solution on functions F7, F8, F15, and F21 with a slow convergence rate, SCHO has a strong ability to converge to the optimal solution in the later iterations compared to other algorithms, which is also seen for functions F22 and F23. It is intuitively recognized from these convergence curves how these three search strategies, including the first phase and second phase of exploration and exploitation and the bounded search strategy, affect the convergence for these functions. The first search strategy plays an important role in F5, F8, F12, F13, F16, F17, F19 and F20-F23, while the second strategy possesses a significant effect on F1-F4 and F9-F11. The second strategy cooperates with the first search strategy to obtain the optimal solutions. For F8, F21–23, the third strategy is also very important and assists the first strategy in helping the algorithm continuously escape from local optima in the later iterations, which also benefits from the fact that the switch mechanism can guarantee exploration in the later stage. In addition, these three strategies have obvious synergy on F7, F14 and F15 to obtain better solutions. Therefore, SCHO can easily escape from the local optima, attributed to the applied three search strategy and good switch mechanism.

Compared to other meta-heuristic optimization algorithms for the 23 classical benchmark functions, SCHO possesses good exploration and exploitation capabilities and strong local optimum avoidance. In conclusion, SCHO is a competitive optimization algorithm.

3.1.4. Scalability analysis of SCHO

To study the effect of optimization problems at high dimensions on the performance of SCHO, a scalability analysis is performed by using 13 classical benchmark functions with dimension levels of 100 and 500. The search agents, iterations and running times of SCHO and comparative algorithms are set to 30, 500 and 30, respectively. Table 9 and Table 10 are the optimization results at $D = 100$ and $D = 500$, respectively. The results show that SCHO still gets the minimum value on Friedman ranking test, which denotes that SCHO is outperform to other algorithms on solving optimization problems at high dimension. Although the stability of SCHO for F2 and F4 at high dimension decreases, SCHO gets better ranking compared with the results for F5, F12 and F13 with $D = 30$. Hence, the value of average ranking of SCHO at high dimension is smaller than the value of 2.1 at low dimension, which indicates that the effect of high dimension on the scalability of SCHO is

Table 7
Optimization results of SCHO and other algorithms for the 23 classical benchmark functions.

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
F1	Best	0.000E+00	3.875E-29	1.873E-04	2.957E-01	2.751E-08	7.080E-151	0.000E+00	1.970E-146	5.285E-59
	Average	0.000E+00	1.625E-27	1.178E-03	2.598E+01	8.930E-07	1.717E-15	0.000E+00	4.022E-141	7.423E-55
	STD	0.000E+00	3.875E-27	1.160E-03	4.588E+01	3.636E-06	9.406E-15	0.000E+00	2.061E-140	1.608E-54
	Rank	1	5	8	9	7	6	1	3	4
F2	Best	0.000E+00	7.045E-18	4.267E-01	1.462E-04	1.345E-01	0.000E+00	0.000E+00	1.988E-81	1.114E-33
	Average	0.000E+00	1.340E-16	4.614E+01	2.364E-02	2.036E+00	0.000E+00	0.000E+00	3.010E-78	1.580E-32
	STD	0.000E+00	1.686E-16	4.465E+01	6.863E-02	1.801E+00	0.000E+00	0.000E+00	7.380E-78	1.657E-32
	Rank	1	6	9	7	8	1	1	4	5
F3	Best	0.000E+00	1.022E-08	1.165E+03	1.275E+03	4.208E+02	3.521E-123	0.000E+00	2.197E-108	1.530E-22
	Average	0.000E+00	2.173E-05	4.713E+03	9.576E+03	1.750E+03	3.803E-03	0.000E+00	2.578E-98	2.718E-16
	STD	0.000E+00	4.713E-05	2.318E+03	5.640E+03	1.307E+03	1.044E-02	0.000E+00	7.946E-98	1.300E-15
	Rank	1	5	8	9	7	6	1	3	4
F4	Best	0.000E+00	4.613E-08	4.977E+00	1.126E+01	6.471E+00	2.591E-56	0.000E+00	1.822E-59	1.979E-18
	Average	0.000E+00	1.207E-06	1.664E+01	3.650E+01	1.175E+01	2.978E-02	0.000E+00	4.991E-57	6.210E-16
	STD	0.000E+00	1.624E-06	5.424E+00	1.444E+01	3.590E+00	2.197E-02	0.000E+00	1.081E-56	1.274E-15
	Rank	1	5	8	9	7	6	1	3	4
F5	Best	2.811E+01	2.606E+01	2.443E+01	4.576E+01	1.876E+01	2.774E+01	1.043E-25	2.718E+01	2.716E+01
	Average	2.888E+01	2.606E+01	3.737E+02	4.027E+04	3.493E+02	2.852E+01	1.542E+01	2.820E+01	2.799E+01
	STD	2.002E-01	7.674E-01	5.330E+02	7.277E+04	4.184E+02	2.714E-01	1.467E+01	6.242E-01	6.407E-01
	Rank	6	2	8	9	7	5	1	4	3
F6	Best	5.376E-05	6.376E-05	2.011E-04	4.353E+00	2.446E-08	2.218E+00	3.410E+00	2.327E+00	1.642E+00
	Average	1.993E+00	8.100E-01	1.117E-03	1.485E+01	1.835E-07	3.126E+00	6.556E+00	3.209E+00	2.606E+00
	STD	1.785E+00	5.047E-01	1.083E-03	1.420E+01	2.353E-07	3.253E-01	1.062E+00	4.943E-01	4.679E-01
	Rank	4	3	2	9	1	6	8	7	5
F7	Best	7.659E-07	2.612E-04	1.142E-01	1.486E-02	8.625E-02	7.258E-07	4.066E-06	1.177E-05	8.960E-05
	Average	6.239E-05	1.893E-03	2.645E-01	1.176E-01	1.677E-01	7.968E-05	9.112E-05	1.251E-04	5.253E-04
	STD	5.154E-05	1.181E-03	9.686E-02	9.062E-02	6.191E-02	7.319E-05	1.086E-04	1.239E-04	4.061E-04
	Rank	1	6	9	7	8	2	3	4	5
F8	Best	−1.255E+04	−8.069E+03	−9.093E+03	−4.446E+03	−8.379E+03	−5.887E+03	−5.667E+03	−7.427E+03	−7.704E+03
	Average	−7.923E+03	−6.116E+03	−5.781E+03	−3.738E+03	−7.475E+03	−5.161E+03	−5.430E+03	−6.306E+03	−3.975E+03
	STD	2.356E+03	1.182E+03	9.714E+02	3.119E+02	6.610E+02	4.131E+02	2.191E+02	6.111E+02	1.350E+03
	Rank	1	4	5	9	2	7	6	3	8
F9	Best	0.000E+00	0.000E+00	4.875E+01	9.072E-02	2.189E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Average	0.000E+00	2.093E+00	8.218E+01	3.463E+01	5.134E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	STD	0.000E+00	2.626E+00	2.083E+01	2.602E+01	1.445E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Rank	1	6	9	7	8	1	1	1	1
F10	Best	4.441E-16	7.505E-14	1.341E+00	2.635E-01	1.842E-01	4.441E-16	4.441E-16	3.997E-15	3.997E-15
	Average	4.441E-16	1.016E-13	4.878E+00	1.296E+01	2.478E+00	4.441E-16	4.441E-16	3.997E-15	6.721E-15
	STD	0.000E+00	1.799E-14	2.736E+00	9.223E+00	7.645E-01	0.000E+00	0.000E+00	0.000E+00	1.528E-15
	Rank	1	6	8	9	7	1	1	4	5
F11	Best	0.000E+00	0.000E+00	2.135E-02	4.354E-02	1.162E-03	2.115E-02	0.000E+00	0.000E+00	0.000E+00
	Average	0.000E+00	5.059E-03	6.136E-02	8.926E-01	1.476E-02	2.336E-01	0.000E+00	0.000E+00	0.000E+00
	STD	0.000E+00	9.184E-03	2.883E-02	3.834E-01	1.005E-02	1.264E-01	0.000E+00	0.000E+00	0.000E+00
	Rank	1	5	7	9	6	8	1	1	1
F12	Best	2.925E-09	6.569E-03	7.820E+00	7.486E-01	2.177E+00	4.038E-01	4.333E-01	1.087E-01	9.619E-02
	Average	2.571E-01	4.048E-02	1.355E+01	2.280E+04	6.455E+00	5.079E-01	1.193E+00	2.550E-01	2.125E-01
	STD	4.358E-01	2.280E-02	5.292E+00	6.398E+04	2.904E+00	4.972E-02	3.268E-01	9.214E-02	1.209E-01
	Rank	4	1	7	9	8	5	6	3	2
F13	Best	3.926E-06	7.429E-02	8.138E-01	2.385E+00	1.211E-02	2.556E+00	1.017E-30	1.171E+00	1.123E+00
	Average	1.594E+00	6.416E-01	2.982E+01	4.235E+05	1.626E+01	2.795E+00	3.307E-01	2.102E+00	1.561E+00
	STD	8.875E-01	2.961E-01	1.724E+01	1.328E+06	1.160E+01	1.086E-01	8.367E-01	3.323E-01	1.963E-01
	Rank	4	2	8	9	7	6	1	5	3
F14	Best	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	1.992E+00	1.072E+00	9.980E-01	9.980E-01
	Average	5.3593	3.5797	2.384	2.1887	1.2297	10.5152	4.0205	4.7771	5.0442
	STD	4.950E+00	3.727E+00	1.525E+00	9.883E-01	5.641E-01	3.539E+00	3.184E+00	4.656E+00	4.335E+00
	Rank	8	4	3	2	1	9	5	6	7
F15	Best	3.075E-04	3.075E-04	5.683E-04	4.574E-04	3.078E-04	3.400E-04	6.476E-04	3.101E-04	3.076E-04
	Average	3.261E-04	4.399E-03	3.879E-03	1.028E-03	4.199E-03	2.101E-02	1.367E-03	3.925E-04	1.754E-03
	STD	1.774E-05	8.121E-03	1.253E-02	3.783E-04	1.167E-02	2.846E-02	5.964E-04	2.391E-04	5.060E-03
	Rank	1	8	6	3	7	9	4	2	5
F16	Best	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00	−1.032E+00
	Average	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0298	−1.0316	−1.0316
	STD	1.648E-08	2.627E-08	1.067E-13	2.848E-05	1.742E-14	1.720E-07	5.715E-03	1.049E-08	2.677E-05
	Rank	1	1	1	1	1	1	9	1	1
F17	Best	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.980E-01	3.984E-01	3.979E-01	3.979E-01
	Average	0.3979	0.3979	0.3979	0.4017	0.3979	0.4095	0.4416	0.3990	0.3980
	STD	1.172E-06	3.637E-06	1.193E-13	4.943E-03	3.365E-14	1.117E-02	1.506E-01	2.270E-03	5.804E-04
	Rank	1	1	1	7	1	8	9	6	5
F18	Best	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00
	Average	6.1545	5.7000	3.0000	3.0001	3.0000	9.3000	3.9187	3.0000	3.0000
	STD	7.360E+00	1.479E+01	7.618E-13	1.211E-04	1.713E-13	1.161E+01	4.964E+00	2.034E-08	1.068E-05
	Rank	8	7	1	5	1	9	6	1	1
F19	Best	−3.863E+00	−3.863E+00	−3.863E+00	−3.862E+00	−3.863E+00	−3.860E+00	−3.858E+00	−3.863E+00	−3.863E+00
	Average	−3.8628	−3.8620	−3.8628	−3.8547	−3.8628	−3.8524	−3.8096	−3.8569	−3.8586

(continued on next page)

Table 7 (*continued*)

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
	STD	3.066E-06	1.973E-03	1.235E-13	2.892E-03	1.073E-09	4.124E-03	3.421E-02	3.794E-03	3.856E-03
	Rank	1	4	1	7	1	8	9	6	5
F20	Best	−3.322E+00	−3.322E+00	−3.322E+00	−3.247E+00	−3.322E+00	−3.225E+00	−3.067E+00	−3.320E+00	−3.322E+00
	Average	−3.2477	−3.2487	−3.2621	−2.7874	−3.2057	−3.0674	−2.586	−2.9753	−3.1389
	STD	7.561E-02	8.560E-02	6.095E-02	5.354E-01	4.071E-02	1.116E-01	4.383E-01	3.055E-01	1.140E-01
	Rank	3	2	1	8	4	6	9	7	5
F21	Best	−1.015E+01	−1.015E+01	−1.015E+01	−6.259E+00	−1.015E+01	−6.269E+00	−5.055E+00	−1.007E+01	−1.015E+01
	Average	−9.2330	−9.1361	−6.1155	−2.2408	−6.2201	−4.1517	−5.0292	−5.3257	−8.0097
	STD	1.916E+00	2.367E+00	2.845E+00	1.866E+00	3.397E+00	9.277E-01	1.423E-01	1.942E+00	3.218E+00
	Rank	1	2	5	9	4	8	7	6	3
F22	Best	−1.040E+01	−1.040E+01	−1.040E+01	−8.515E+00	−1.040E+01	−8.812E+00	−5.088E+00	−1.036E+01	−1.040E+01
	Average	−9.0825	−10.2254	−6.6803	−3.2226	−7.9169	−3.5547	−5.0877	−5.4462	−9.2524
	STD	2.521E+00	9.629E-01	3.192E+00	1.975E+00	3.404E+00	1.856E+00	8.450E-07	2.684E+00	2.347E+00
	Rank	3	1	5	9	4	8	7	6	2
F23	Best	−1.054E+01	−1.054E+01	−1.054E+01	−8.394E+00	−1.054E+01	−6.790E+00	−5.129E+00	−1.045E+01	−1.054E+01
	Average	−8.6483	−10.2641	−6.3772	−4.4546	−7.963	−3.7526	−5.1285	−5.6363	−10.0737
	STD	2.949E+00	1.481E+00	3.323E+00	1.686E+00	3.718E+00	1.447E+00	2.004E-06	1.920E+00	1.745E+00
	Rank	3	1	5	8	4	9	7	6	2
Mean	Rank	2.48	3.78	5.43	7.39	4.83	5.83	4.57	4.00	3.74
Final	Ranking	1	3	7	9	6	8	5	4	2

Table 8
Wilcoxon rank sum test between SCHO and other algorithms for the 23 classical benchmark functions with $D = 30$, except for fixed-dimension functions (F14–F23).

F	GWO p-value	Sig.	ALO p-value	Sig.	SCA p-value	Sig.	SSA p-value	Sig.	AOA p-value	Sig.	RSA p-value	Sig.	SHO p-value	Sig.	GJO p-value	Sig.
F1	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F2	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	1.2E-12	+	1.2E-12	+
F3	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F4	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F5	1.0E-08	−	2.2E-03	+	3.0E-11	+	5.1E-08	+	4.6E-03	−	2.2E-01	~	9.6E-02	~	7.7E-04	−
F6	2.8E-02	−	2.2E-03	−	3.0E-11	+	3.0E-11	+	1.8E-10	+	3.3E-11	+	1.4E-09	+	2.6E-07	+
F7	3.0E-11	+	2.2E-03	+	3.0E-11	+	3.0E-11	−	2.0E-01	+	1.3E-01	~	1.8E-01	~	4.0E-08	+
F8	2.4E-03	+	1.5E-02	+	3.3E-11	+	3.4E-04	+	7.2E-05	+	3.8E-03	+	2.9E-03	+	5.0E-04	+
F9	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	NaN	~	NaN	~
F10	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	7.2E-13	+	2.0E-13	+
F11	5.6E-03	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	NaN	~
F12	4.0E-04	−	2.2E-03	+	3.0E-11	+	3.0E-11	+	1.6E-08	−	6.7E-11	+	1.0E-04	~	3.2E-04	−
F13	1.2E-03	−	2.6E-02	+	3.0E-11	+	8.9E-06	+	4.5E-11	+	5.6E-07	−	6.4E-03	−	5.4E-01	~
F14	9.3E-01	~	6.9E-01	~	1.3E-02	−	3.0E-07	−	1.2E-06	+	8.2E-02	~	7.6E-01	~	1.7E-01	~
F15	1.3E-02	+	2.2E-03	+	4.1E-11	+	3.0E-11	+	8.2E-11	+	3.0E-11	+	1.6E-01	~	1.3E-03	+
F16	1.5E-01	~	2.2E-03	−	3.0E-11	+	3.0E-11	+	8.5E-09	+	3.0E-11	+	1.9E-07	−	7.7E-08	+
F17	2.3E-01	~	2.2E-03	−	3.0E-11	+	3.0E-11	−	3.0E-11	+	3.0E-11	+	1.1E-04	+	7.6E-07	+
F18	2.6E-03	−	2.2E-03	−	1.2E-04	−	3.0E-11	−	2.9E-01	+	2.1E-03	−	3.7E-11	−	1.6E-02	−
F19	7.1E-08	+	2.2E-03	−	4.1E-11	+	3.0E-11	−	4.5E-11	+	3.0E-11	+	4.8E-07	+	8.2E-07	+
F20	9.0E-04	−	8.7E-03	−	3.3E-11	+	5.3E-03	+	2.0E-10	+	5.5E-11	+	8.4E-08	+	2.2E-01	+
F21	1.1E-03	+	6.5E-02	~	8.2E-11	+	2.1E-03	+	8.2E-11	+	8.5E-09	+	3.5E-09	+	2.7E-02	+
F22	1.6E-03	−	6.5E-02	~	7.1E-08	+	3.1E-02	+	1.2E-05	+	8.5E-09	+	1.3E-09	+	5.6E-01	~
F23	1.3E-02	−	2.2E-03	+	6.0E-09	+	3.4E-02	+	8.5E-09	+	9.5E-06	+	8.1E-05	+	6.7E-01	~
(W|L|T)	(12|8|3)		(14|6|3)		(21|2|0)		(18|5|0)		(18|2|3)		(11|2|10)		(13|3|7)		(14|3|6)	

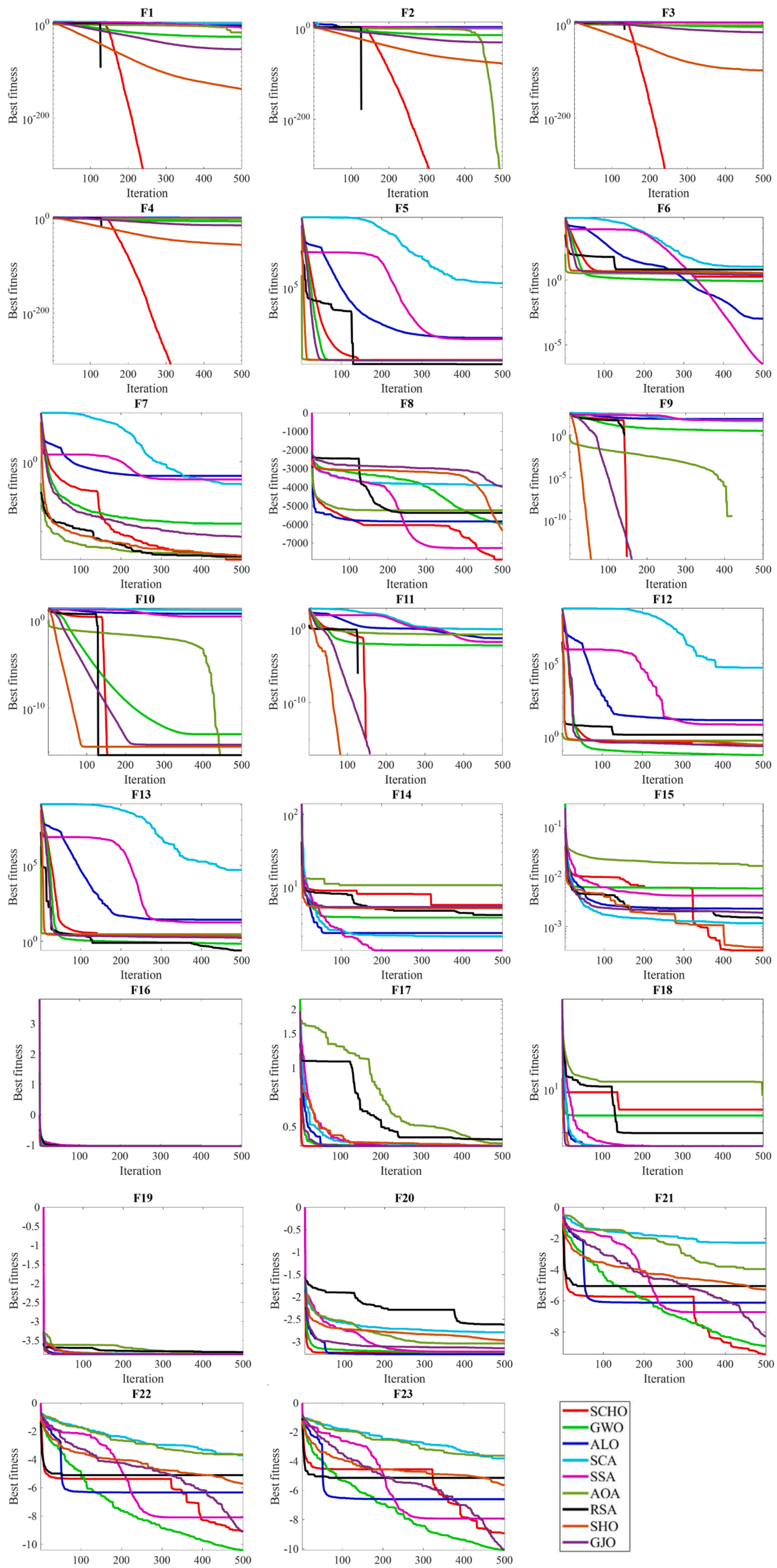


Fig. 9. Convergence curves of SCHO and comparison algorithms for 23 classical benchmark functions.

Table 9
Optimization results of SCHO and other algorithms for the 13 classical benchmark functions (F1-F13) with fixed dimension ($D = 100$).

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
F1	Best	0.000E+00	2.967E-13	2.725E+03	1.163E+03	6.251E+02	7.818E-03	0.000E+00	5.515E-125	1.080E-29
	Average	0.000E+00	1.562E-12	4.243E+03	9.381E+03	1.377E+03	2.738E-02	0.000E+00	2.453E-120	9.724E-28
	STD	0.000E+00	1.229E-12	1.520E+03	6.695E+03	4.002E+02	1.134E-02	0.000E+00	5.579E-120	1.637E-27
	Rank	1	5	8	9	7	6	1	3	4
F2	Best	0.000E+00	2.261E-08	8.204E+01	4.937E-01	3.364E+01	4.298E-132	0.000E+00	4.655E-68	2.263E-18
	Average	7.576E-263	4.360E-08	8.347E+10	7.918E+00	4.544E+01	3.628E-62	0.000E+00	1.267E-65	1.106E-17
	STD	0.000E+00	1.934E-08	2.614E+11	6.114E+00	5.039E+00	1.987E-61	0.000E+00	4.836E-65	8.101E-18
	Rank	2	6	9	7	8	4	1	3	5
F3	Best	0.000E+00	4.650E+00	6.425E+04	1.524E+05	1.483E+04	2.228E-01	0.000E+00	5.713E-92	8.545E-06
	Average	0.000E+00	5.781E+02	8.982E+04	2.278E+05	4.754E+04	8.509E-01	0.000E+00	1.559E-83	9.311E+00
	STD	0.000E+00	6.949E+02	2.244E+04	4.390E+04	2.531E+04	7.137E-01	0.000E+00	8.287E-83	5.047E+01
	Rank	1	6	8	9	7	4	1	3	5
F4	Best	0.000E+00	2.678E-02	2.727E+01	7.673E+01	2.190E+01	7.685E-02	0.000E+00	1.318E-52	3.169E-04
	Average	4.243E-249	6.978E-01	3.815E+01	8.971E+01	2.741E+01	9.564E-02	0.000E+00	6.906E-50	2.265E+00
	STD	0.000E+00	6.368E-01	7.427E+00	3.247E+00	3.425E+00	1.286E-02	0.000E+00	1.117E-49	4.141E+00
	Rank	2	5	8	9	7	4	1	3	6
F5	Best	9.875E+01	9.589E+01	1.537E+05	2.056E+07	5.850E+04	9.826E+01	9.877E+01	9.795E+01	9.718E+01
	Average	9.887E+01	9.796E+01	6.031E+05	1.113E+08	1.692E+05	9.884E+01	9.898E+01	9.858E+01	9.838E+01
	STD	1.004E-01	7.589E-01	4.238E+05	4.437E+07	8.706E+04	1.490E-01	4.062E-02	2.818E-01	4.492E-01
	Rank	5	1	8	9	7	4	6	3	2
F6	Best	1.466E-04	8.441E+00	2.747E+03	1.474E+03	7.308E+02	1.675E+01	2.443E+01	1.673E+01	1.474E+01
	Average	8.562E+00	9.799E+00	4.423E+03	1.041E+04	1.446E+03	1.823E+01	2.469E+01	1.847E+01	1.642E+01
	STD	8.137E+00	9.496E-01	1.598E+03	7.475E+03	3.772E+02	6.265E-01	1.245E-01	8.060E-01	9.068E-01
	Rank	1	2	8	9	7	4	6	5	3
F7	Best	5.429E-07	3.274E-03	4.537E+00	3.106E+01	1.822E+00	5.753E-06	1.834E-07	1.613E-05	2.415E-04
	Average	7.285E-05	6.890E-03	5.971E+00	1.430E+02	2.842E+00	8.398E-05	8.433E-05	1.352E-04	1.348E-03
	STD	5.388E-05	1.994E-03	2.234E+00	7.330E+01	5.130E-01	7.120E-05	7.447E-05	9.588E-05	9.072E-04
	Rank	1	6	8	9	7	2	3	4	5
F8	Best	−3.926E+04	−2.073E+04	−1.806E+04	−8.409E+03	−2.369E+04	−1.145E+04	−1.806E+04	−1.220E+04	−1.653E+04
	Average	−1.856E+04	−1.635E+04	−1.806E+04	−6.972E+03	−2.057E+04	−1.007E+04	−1.705E+04	−1.021E+04	−8.584E+03
	STD	9.867E+03	1.343E+03	3.835E-12	5.866E+02	1.739E+03	7.221E+02	7.228E+02	8.757E+02	3.864E+03
	Rank	2	5	3	9	1	7	4	6	8
F9	Best	0.000E+00	1.587E-10	3.125E+02	8.865E+01	1.586E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Average	0.000E+00	1.011E+01	3.535E+02	2.581E+02	2.474E+02	0.000E+00	0.000E+00	0.000E+00	3.790E-15
	STD	0.000E+00	6.742E+00	3.724E+01	1.220E+02	3.880E+01	0.000E+00	0.000E+00	0.000E+00	2.076E-14
	Rank	1	6	9	8	7	1	1	1	5
F10	Best	4.441E-16	4.355E-08	1.364E+01	4.146E+00	7.981E+00	4.441E-16	4.441E-16	4.441E-16	3.952E-14
	Average	4.441E-16	1.110E-07	1.440E+01	1.908E+01	1.025E+01	6.139E-04	4.441E-16	3.878E-15	5.089E-14
	STD	0.000E+00	4.695E-08	7.358E-01	4.142E+00	1.067E+00	1.052E-03	0.000E+00	6.486E-16	1.007E-14
	Rank	1	5	8	9	7	6	1	3	4
F11	Best	0.000E+00	5.995E-14	2.886E+01	1.556E+01	7.278E+00	1.969E+02	0.000E+00	0.000E+00	0.000E+00
	Average	0.000E+00	2.277E-03	3.489E+01	9.554E+01	1.222E+01	5.921E+02	0.000E+00	0.000E+00	0.000E+00
	STD	0.000E+00	7.018E-03	5.162E+00	6.075E+01	3.085E+00	1.958E+02	0.000E+00	0.000E+00	0.000E+00
	Rank	1	5	7	8	6	9	1	1	1
F12	Best	9.093E-10	1.721E-01	6.599E+01	5.477E+07	1.727E+01	8.733E-01	9.107E-01	5.931E-01	4.814E-01
	Average	6.876E-01	2.832E-01	1.386E+03	3.216E+08	3.707E+01	9.135E-01	1.256E+00	7.022E-01	6.119E-01
	STD	6.422E-01	6.849E-02	2.621E+03	1.596E+08	1.250E+01	1.987E-02	8.248E-02	8.099E-02	7.777E-02
	Rank	3	1	8	9	7	5	6	4	2
F13	Best	1.024E+00	6.057E+00	6.666E+03	7.741E+07	1.713E+02	9.868E+00	6.266E+00	9.364E+00	7.620E+00
	Average	9.580E+00	7.035E+00	1.046E+05	5.066E+08	4.372E+03	9.985E+00	9.711E+00	9.654E+00	8.445E+00
	STD	1.616E+00	4.309E-01	1.054E+05	2.481E+08	6.148E+03	4.180E-02	6.573E-01	1.384E-01	3.438E-01
	Rank	3	1	8	9	7	6	5	4	2
Mean	Rank	1.85	4.15	7.69	8.69	6.54	4.77	2.85	3.31	4.00
Final	Ranking	1	5	8	9	7	6	2	3	4

lower than other algorithms. Table 11 and Table 12 report the results of Wilcoxon rank sum test between SCHO and other algorithms for the 13 classical benchmark test functions at $D = 100$ and $D = 500$, respectively. It can be seen that SCHO performs better than the other algorithms for most functions, especially GWO, ALO, SCA, SSA and AOA. Therefore, SCHO still possesses better optimization ability compared to other algorithms for solving optimization problems at high dimensions.

3.2. Optimization results of SCHO for CEC 2014 test functions

In this section, a more complex set of CEC 2014 test functions [111] is selected to further evaluate the optimization capabilities of SCHO, as shown in Table 13. These functions include unimodal functions, simple multimodal functions, hybrid functions and composition functions, in which hybrid functions and composition functions account for 50%. SCHO and other algorithms including GWO, ALO, SCA, SSA, AOA, RSA, SHO and GJO use the same parameters as before for CEC 2014 test functions except for the dimension of CEC 2014, which is set to 10.

The optimization results from 30 runs are shown in Table 14, including best fitness, average fitness, STD and Friedman ranking test. It can be observed that SCHO obtains the minimum value in Friedman ranking test, followed by GWO and SSA, which denotes SCHO is a competitive algorithm compared to other algorithms. In addition, the results also show that SCHO presents the best optimization ability for solving composition functions (F23-F30), with 75% of the optimal values (F23, F25, F27-F30), which also indicates that SCHO possesses superior performance for complex problems. The results of Wilcoxon rank sum test between SCHO and other algorithms for CEC 2014 test functions ($\alpha = 0.05$, $D = 10$) are shown in Table 15. The results demonstrate that SCHO performs better than the other algorithms for most functions, especially SCA, AOA and RSA. Also, SCHO shows a superior performance on F23-F30 compared with other algorithms, which

Table 10
Optimization results of SCHO and comparison algorithms on 13 classical benchmark functions (F1-F13) with fixed dimension($D = 500$).

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
F1	Best	0.000E+00	8.358E-04	1.778E+05	2.030E+04	8.615E+04	5.553E-01	0.000E+00	5.746E-115	7.198E-14
	Average	0.000E+00	1.523E-03	2.042E+05	1.859E+05	9.355E+04	6.479E-01	0.000E+00	1.839E-108	7.464E-13
	STD	0.000E+00	5.304E-04	3.555E+04	8.378E+04	5.797E+03	4.159E-02	0.000E+00	8.196E-108	7.954E-13
	Rank	1	5	9	8	7	6	1	3	4
F2	Best	0.000E+00	7.251E-03	2.256E+03	5.238E+01	5.017E+02	1.290E-20	0.000E+00	3.255E-63	2.872E-09
	Average	3.513E-211	1.065E-02	3.682E+211	1.106E+02	5.363E+02	6.151E-04	0.000E+00	3.478E-59	6.277E-09
	STD	0.000E+00	2.156E-03	Inf	4.695E+01	2.108E+01	9.417E-04	0.000E+00	1.302E-58	2.710E-09
	Rank	2	6	9	7	8	5	1	3	4
F3	Best	0.000E+00	1.771E+05	1.536E+06	4.986E+06	6.395E+05	1.742E+01	0.000E+00	1.452E-82	4.149E+03
	Average	0.000E+00	3.198E+05	2.082E+06	7.149E+06	1.364E+06	3.900E+01	0.000E+00	1.876E-74	6.554E+04
	STD	0.000E+00	6.643E+04	4.914E+05	7.149E+06	5.778E+05	1.658E+01	0.000E+00	1.023E-73	7.233E+04
	Rank	1	6	8	9	7	4	1	3	5
F4	Best	0.000E+00	5.047E+01	4.420E+01	9.821E+01	3.484E+01	1.580E-01	0.000E+00	1.687E-48	7.540E+01
	Average	1.492E-204	6.445E+01	5.203E+01	9.905E+01	3.989E+01	1.785E-01	0.000E+00	8.057E-46	8.193E+01
	STD	0.000E+00	4.594E+00	9.203E+00	3.174E-01	2.638E+00	1.611E-02	0.000E+00	1.891E-45	3.662E+00
	Rank	2	7	6	9	5	4	1	3	8
F5	Best	4.988E+02	4.973E+02	5.786E+07	1.001E+09	2.542E+07	4.989E+02	4.990E+02	4.979E+02	4.979E+02
	Average	4.990E+02	4.980E+02	1.149E+08	1.964E+09	3.657E+07	4.991E+02	4.990E+02	4.986E+02	4.983E+02
	STD	5.200E-02	2.628E-01	4.121E+07	4.967E+08	5.727E+06	8.129E-02	1.044E-03	1.543E-01	1.172E-01
	Rank	4	1	8	9	7	6	5	3	2
F6	Best	1.242E+02	8.809E+01	1.872E+05	7.069E+04	8.013E+04	1.139E+02	1.244E+02	1.148E+02	1.073E+02
	Average	1.246E+02	9.163E+01	2.084E+05	2.026E+05	9.259E+04	1.160E+02	1.247E+02	1.169E+02	1.093E+02
	STD	1.481E-01	2.104E+00	2.165E+04	7.158E+04	6.732E+03	1.183E+00	9.114E-02	7.992E-01	1.085E+00
	Rank	5	1	9	8	7	3	6	4	2
F7	Best	4.091E-06	3.343E-02	4.991E+02	8.698E+03	1.925E+02	1.550E-05	2.709E-06	2.697E-05	1.550E-03
	Average	8.216E-05	4.816E-02	7.993E+02	1.533E+04	2.753E+02	1.043E-04	9.321E-05	1.358E-04	6.605E-03
	STD	6.352E-05	1.253E-02	3.213E+02	2.962E+03	3.722E+01	7.519E-05	9.799E-05	9.587E-05	3.598E-03
	Rank	1	6	8	9	7	3	2	4	5
F8	Best	−1.944E+05	−7.113E+04	−1.274E+05	−1.731E+04	−6.843E+04	−2.697E+04	−7.088E+04	−2.741E+04	−4.937E+04
	Average	−6.282E+04	−5.537E+04	−9.232E+04	−1.484E+04	−5.980E+04	−2.252E+04	−6.145E+04	−2.233E+04	−2.544E+04
	STD	5.105E+04	1.175E+04	7.893E+03	9.781E+02	4.690E+03	1.477E+03	6.012E+03	1.889E+03	1.533E+04
	Rank	2	5	1	6	4	8	3	9	7
F9	Best	0.000E+00	2.844E+01	3.555E+03	4.545E+02	2.955E+03	0.000E+00	0.000E+00	0.000E+00	3.638E-12
	Average	0.000E+00	7.023E+01	3.860E+03	1.165E+03	3.159E+03	6.826E-06	0.000E+00	0.000E+00	6.852E-12
	STD	0.000E+00	2.692E+01	2.007E+02	4.574E+02	1.276E+02	6.483E-06	0.000E+00	0.000E+00	2.186E-12
	Rank	1	6	9	7	8	5	1	1	4
F10	Best	4.441E-16	1.350E-03	1.557E+01	9.310E+00	1.361E+01	7.029E-03	4.441E-16	3.997E-15	9.452E-09
	Average	4.441E-16	1.882E-03	1.640E+01	1.906E+01	1.417E+01	7.909E-03	4.441E-16	4.115E-15	3.065E-08
	STD	0.000E+00	4.270E-04	7.167E-01	3.643E+00	2.568E-01	3.505E-04	0.000E+00	6.486E-16	1.280E-08
	Rank	1	5	8	9	7	6	1	3	4
F11	Best	0.000E+00	6.110E-05	1.474E+03	6.561E+02	7.473E+02	6.545E+03	0.000E+00	0.000E+00	9.659E-15
	Average	0.000E+00	2.231E-02	2.109E+03	1.824E+03	8.491E+02	9.606E+03	0.000E+00	0.000E+00	8.888E-14
	STD	0.000E+00	4.102E-02	5.313E+02	5.739E+02	4.823E+01	2.784E+03	0.000E+00	0.000E+00	9.829E-14
	Rank	1	5	7	8	6	9	1	1	4
F12	Best	9.540E-08	6.762E-01	4.069E+07	3.951E+09	1.909E+05	1.052E+00	1.192E+00	1.002E+00	8.661E-01
	Average	1.040E+00	7.768E-01	5.490E+07	6.110E+09	1.311E+06	1.085E+00	1.202E+00	1.052E+00	9.358E-01
	STD	4.150E-01	5.209E-02	1.433E+07	1.056E+09	7.762E+05	1.143E-02	3.540E-03	1.864E-02	2.247E-02
	Rank	3	1	8	9	7	5	6	4	2
F13	Best	2.821E+01	4.863E+01	1.234E+08	5.376E+09	2.466E+07	5.012E+01	4.990E+01	4.941E+01	4.722E+01
	Average	4.916E+01	5.065E+01	2.071E+08	9.721E+09	3.588E+07	5.020E+01	4.992E+01	4.963E+01	4.800E+01
	STD	3.957E+00	1.483E+00	8.615E+07	2.473E+09	8.533E+06	3.525E-02	3.492E-02	9.757E-02	3.433E-01
	Rank	2	6	8	9	7	5	4	3	1
Mean	Rank	2.00	4.62	7.54	8.23	6.69	5.31	2.54	3.38	4.00
Final	Ranking	1	5	8	9	7	6	2	3	4

is also consistent with the results of the Friedman ranking test. Hence, SCHO still presents excellent performance to find solutions for solving complex problems.

3.3. SCHO for optimizing engineering design problems

To evaluate the efficiency of SCHO, six well-known engineering design problems with several inequality constraints are performed, including tension/compression spring design problem, pressure vessel design problem, welded beam design problem, speed reducer design problem, cantilever beam design problem, and three-bar truss design problem. A simple death penalty is applied to solve the constraints of these engineering problems. When the search agent is not within the scope of any constraints, its fitness will be given a significant value. Then, it will be discarded in the next iteration, which can change the constrained design problems into unconstrained ones to look for the global optimal solution.

3.3.1. Tension/compression spring design problem

The aim of the tension/compression spring design problem, shown in Fig. 10, is to minimize the overall weight by selecting suitable parameter values containing wire diameter (d), mean coil diameter (D), and the number of active coils (P). The mathematical formulation of this engineering design problem is as follows.

Given the variables: $x_1 = d,\ x_2 = D,\ x_3 = P$

Minimize:

$$f(x) = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1(x) = 1 - \frac{x_3x_2^3}{71785x_1^4} \leq 0,$$

Table 11
Wilcoxon rank sum test between SCHO and the other algorithms for the 13 classical benchmark test functions at $\alpha = 0.05$ and $D = 100$.

F	GWO		ALO		SCA		SSA		AOA		RSA		SHO		GJO	
	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.
F1	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F2	2.3E-11	+	2.2E-03	+	2.3E-11	+	2.3E-11	+	2.3E-11	+	6.6E-05	–	2.3E-11	+	2.3E-11	+
F3	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F4	5.2E-12	+	2.2E-03	+	5.2E-12	+	5.2E-12	+	5.2E-12	+	2.9E-05	–	5.2E-12	+	5.2E-12	+
F5	3.0E-11	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	1.3E-02	–	3.1E-08	+	3.0E-11	–	3.0E-11	–
F6	4.2E-02	+	2.2E-03	+	3.0E-11	+	3.0E-11	+	4.0E-03	+	3.0E-11	+	6.7E-03	+	8.0E-01	+
F7	3.0E-11	+	2.2E-03	+	3.0E-11	+	3.0E-11	+	4.9E-01	~	6.6E-02	~	4.1E-02	+	7.4E-11	+
F8	2.3E-03	+	3.5E-02	+	3.0E-11	+	5.2E-07	–	4.2E-09	+	3.2E-03	+	3.0E-11	+	2.4E-05	+
F9	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	NaN	~	3.3E-01	+
F10	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	2.8E-03	+	NaN	~	1.2E-13	+	9.7E-13	+
F11	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	NaN	~
F12	8.0E-03	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	5.2E-02	~	3.8E-03	+	8.7E-03	+	9.1E-03	–
F13	3.0E-11	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	1.4E-09	+	6.7E-06	+	3.0E-11	+	3.0E-11	–
(W|L|T)	(10|3|0)		(13|0|0)		(13|0|0)		(12|1|0)		(9|1|3)		(5|2|6)		(10|1|2)		(9|3|1)	

Table 12
Wilcoxon rank sum test between SCHO and the other algorithms on 13 classical benchmark test functions at $\alpha = 0.05$, $D = 500$.

F	GWO		ALO		SCA		SSA		AOA		RSA		SHO		GJO	
	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.	p-value	Sig.
F1	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F2	2.0E-11	+	2.2E-03	+	2.0E-11	+	2.0E-11	+	2.0E-11	+	2.9E-05	–	2.0E-11	+	2.0E-11	+
F3	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	1.2E-12	+	1.2E-12	+
F4	2.1E-11	+	2.2E-03	+	2.1E-11	+	2.1E-11	+	2.1E-11	+	1.3E-05	–	2.1E-11	+	2.1E-11	+
F5	3.0E-11	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	3.0E-11	+	9.8E-10	+	3.0E-11	–	3.0E-11	–
F6	3.0E-11	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	3.0E-11	–	1.2E-02	+	3.0E-11	–	3.0E-11	–
F7	3.0E-11	+	2.2E-03	+	3.0E-11	+	3.0E-11	+	7.5E-02	~	8.0E-01	~	1.2E-02	+	3.0E-11	+
F8	4.2E-03	+	4.8E-02	–	3.0E-11	+	3.6E-05	+	1.1E-07	+	5.9E-06	+	1.4E-07	+	1.0E-03	+
F9	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.3E-07	+	NaN	~	NaN	~	1.0E-12	+
F10	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	2.7E-14	+	1.2E-12	+
F11	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	1.2E-12	+	NaN	~	NaN	~	1.2E-12	+
F12	1.1E-06	–	2.2E-03	+	3.0E-11	+	3.0E-11	+	1.1E-06	+	3.3E-01	+	1.1E-06	+	1.1E-06	–
F13	1.8E-02	+	2.2E-03	+	3.0E-11	+	3.0E-11	+	3.0E-11	+	6.2E-04	+	1.3E-09	+	5.6E-10	–
(W|L|T)	(10|3|0)		(12|1|0)		(13|0|0)		(13|0|0)		(11|1|1)		(5|2|6)		(9|2|2)		(9|4|0)	

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \le 0,$$

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

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

Variables range:

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

The above formulation shows that the tension/compression spring

Table 13
CEC 2014 test functions.

Typology	Function	Description	f_{min}
Unimodal Functions	F1	Rotated High Conditioned Elliptic Function	100
	F2	Rotated Bent Cigar Function	200
	F3	Rotated Discus Function	300
Simple Multimodal Functions	F4	Shifted and Rotated Rosenbrock's Function	400
	F5	Shifted and Rotated Ackley's Function	500
	F6	Shifted and Rotated Weierstrass Function	600
	F7	Shifted and Rotated Griewank's Function	700
	F8	Shifted Rastrigin's Function	800
	F9	Shifted and Rotated Rastrigin's Function	900
	F10	Shifted Schwefel's Function	1000
	F11	Shifted and Rotated Schwefel's Function	1100
	F12	Shifted and Rotated Katsuura Function	1200
	F13	Shifted and Rotated HappyCat Function	1300
	F14	Shifted and Rotated HGBat Function	1400
	F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
Hybrid Functions	F16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
	F17	Hybrid Function 1 ($N = 3$)	1700
	F18	Hybrid Function 2 ($N = 3$)	1800
	F19	Hybrid Function 3 ($N = 4$)	1900
	F20	Hybrid Function 4 ($N = 4$)	2000
	F21	Hybrid Function 5 ($N = 5$)	2100
	F22	Composition Function 1 ($N = 5$)	2200
Composition Functions	F23	Composition Function 1 ($N = 5$)	2300
	F24	Composition Function 2 ($N = 3$)	2400
	F25	Composition Function 3 ($N = 3$)	2500
	F26	Composition Function 4 ($N = 5$)	2600
	F27	Composition Function 5 ($N = 5$)	2700
	F28	Composition Function 6 ($N = 5$)	2800
	F29	Composition Function 7 ($N = 3$)	2900
	F30	Composition Function 8 ($N = 3$)	3000

design problem considers four constraints. Table 16 lists the results of optimum weight and corresponding variables obtained by SCHO and the 11 meta-heuristic optimization algorithms published in the literature. It can be observed from Table 16 that SCHO gets the optimal solution compared to other algorithms. In detail, the optimum weight obtained by SCHO is 0.0126656, and the values of the corresponding variables x_1, x_2 and x_3 are 0.0517422, 0.3579972, and 11.2146238, respectively. Therefore, the SCHO outperforms all other optimization algorithms for the tension/compression spring design problem.

3.3.2. *Pressure vessel design problem*

One of the most common problems in mixed-integer designs is the pressure vessel design problem. SCHO is applied to look for the minimum cost. Fig. 11 shows that there are four variables in this problem, including the inner radius (R), the shell thickness (T_s), the head thickness (T_h) and the length without the head (L). The mathematical formulation of the pressure vessel design problem is as follows.

Given the variables: $x_1 = T_s,\ x_2 = T_h,\ x_3 = R,\ x_4 = L$

Minimize:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

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

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

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0,$$

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

Variables range:

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

The above formulation shows that four constraints in this engineering problem must be considered for the optimal solution. Table 17 shows the optimization results obtained by SCHO and 11 meta-heuristic optimization algorithms reported in the literature. It can be seen from Table 17 that SCHO obtains the optimum cost of 5889.0061 and its corresponding variables x_1, x_2, x_3 and x_4 are 0.7796836, 0.3854124, 40.39092, and 199.0132, respectively, while the results obtained by other algorithms are more than 6000. It is clear that SCHO is far better than other algorithms in solving the pressure vessel design problem.

3.3.3. *Welded beam design problem*

Minimizing the entire cost of the welded beam is the goal of this engineering design problem shown in Fig. 12. To get the optimum cost, its four variables containing 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) should be well designed, and these variables need to be met seven constraints such as shear stress (τ), the deflection (δ), bending stress in beam (θ) and the bar's buckling load (P_c). The mathematical formulation of the welded beam design problem is as follows.

Given the variables: $x_1 = h,\ x_2 = l,\ x_3 = t,\ x_4 = b$

Minimize:

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

Subject to:

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

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

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

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

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

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

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

Where $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{P}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$, $M = P\left(L + \frac{x_2}{2}\right)$, $R = \sqrt{\frac{x_2^2}{4} + \frac{(x_1+x_3)^2}{2}}$, $J = 2\left[\sqrt{2}x_1x_2\left\{\frac{x_2^2}{4}\right\} + \left(\frac{x_1+x_3}{2}\right)^2\right]$, $\sigma(x) = \frac{6PL}{x_4x_3^2}$, $\delta(x) = \frac{4PL}{Ex_3^3x_4}$, $P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$, $P = 6000\ lb$, $L = 14\ in$, $E = 30 \times 1^6\ psi$, $G = 12 \times 10^6\ psi$, $\tau_{max} = 13,600\ psi$, $\sigma_{max} = 30,000\ psi$, $\delta_{max} = 0.25\ in$

Variables range:

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

The optimization results obtained by SCHO and eleven meta-heuristic algorithms reported in literature are shown in Table 18. It can be seen that SCHO found the optimum cost of 1.72516 compared to the other algorithms. Corresponding parameters x_1, x_2, x_3, x_4 are found to be equal to 0.20565, 3.47312, 9.03685, and 0.20573 respectively. Thence, SCHO is the better choice for solving the welded beam design problem.

3.3.4. *Speed reducer design problem*

Minimizing the total weight of the speed reducer is the main objective of this design problem shown in Fig. 13. It is a more complicated design problem than others since it includes seven design variables and eleven constraints. The design variables include the face width (x_1), the module of teeth (x_2), the number of teeth on the pinion (x_3), the length of the first shaft between bearings (x_4), the length of the second shaft

Table 14
Optimization results of SCHO and other algorithms for CEC 2014 test functions with fixed dimension ($D = 10$).

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
F1	Best	1.1278E+06	8.7768E+05	8.7404E+04	3.1196E+06	2.1759E+05	1.0023E+07	1.9842E+07	3.9498E+06	9.7340E+05
	Average	9.7772E+06	1.0634E+07	1.3071E+06	1.0783E+07	2.9747E+06	7.1376E+07	1.4399E+08	1.2505E+07	1.0342E+07
	STD	1.0876E+07	7.4558E+06	1.1346E+06	4.5463E+06	2.3781E+06	7.5185E+07	1.3184E+08	5.3000E+06	7.7517E+06
	Rank	3	5	1	6	2	8	9	7	4
F2	Best	1.4696E+06	1.3200E+04	2.0316E+02	3.3767E+08	2.1446E+02	4.3629E+09	2.1550E+09	2.6738E+06	1.2324E+05
	Average	2.6184E+08	1.1215E+08	2.9711E+03	9.7474E+08	2.8848E+03	7.0183E+09	5.9787E+09	5.9266E+08	7.1230E+08
	STD	4.1319E+08	3.1847E+08	2.7310E+03	4.0522E+08	3.0505E+03	2.0883E+09	1.5982E+09	6.0406E+08	8.0800E+08
	Rank	4	3	2	7	1	9	8	5	6
F3	Best	1.4712E+03	1.3637E+03	9.8011E+03	3.2364E+03	4.8567E+03	1.2007E+04	4.4842E+03	1.5832E+03	1.6593E+03
	Average	1.1682E+04	1.4029E+04	4.5439E+04	1.2611E+04	1.3880E+04	2.1694E+04	1.2067E+04	5.9377E+03	6.7862E+03
	STD	6.4032E+03	7.3119E+03	2.3479E+04	7.6870E+03	7.0631E+03	6.0569E+03	3.6071E+03	4.8039E+03	3.4714E+03
	Rank	3	7	9	5	6	8	4	1	2
F4	Best	4.0469E+02	4.0907E+02	4.0043E+02	4.4148E+02	4.0434E+02	6.5618E+02	5.8829E+02	4.1332E+02	4.0741E+02
	Average	4.3764E+02	4.4870E+02	4.3983E+02	4.7756E+02	4.3785E+02	1.8332E+03	1.7331E+03	4.6610E+02	4.4345E+02
	STD	2.8354E+01	8.0879E+01	1.8840E+01	2.4195E+01	1.4961E+01	7.7444E+02	9.3367E+02	4.6133E+01	2.6459E+01
	Rank	1	5	3	7	2	9	8	6	4
F5	Best	5.2028E+02	5.2028E+02	5.2000E+02	5.2029E+02	5.2000E+02	5.2006E+02	5.2034E+02	5.2002E+02	5.2016E+02
	Average	5.2047E+02	5.2053E+02	5.2003E+02	5.2050E+02	5.2007E+02	5.2018E+02	5.2050E+02	5.2014E+02	5.2049E+02
	STD	1.0998E-01	1.0816E-01	6.7755E-02	9.6875E-02	1.1229E-01	6.3119E-02	7.9119E-02	7.8075E-02	1.1941E-01
	Rank	6	9	1	8	2	5	9	4	7
F6	Best	6.0147E+02	6.0043E+02	6.0074E+02	6.0509E+02	6.0005E+02	6.0771E+02	6.0880E+02	6.0342E+02	6.0156E+02
	Average	6.0521E+02	6.0293E+02	6.0524E+02	6.0759E+02	6.0498E+02	6.0963E+02	6.1008E+02	6.0604E+02	6.0433E+02
	STD	1.8743E+00	1.3476E+00	1.6985E+00	1.3200E+00	1.8679E+00	9.8641E-01	5.9071E-01	1.4442E+00	1.5259E+00
	Rank	4	1	5	7	3	8	9	6	2
F7	Best	7.0091E+02	7.0032E+02	7.0007E+02	7.0588E+02	7.0007E+02	7.3866E+02	7.3448E+02	7.0096E+02	7.0128E+02
	Average	7.0848E+02	7.0220E+02	7.0023E+02	7.1476E+02	7.0021E+02	8.4081E+02	8.0320E+02	7.1250E+02	7.0999E+02
	STD	1.2273E+01	5.2913E+00	1.0819E-01	4.2924E+00	1.3753E-01	4.2992E+01	3.3119E+01	1.6978E+01	1.1731E+01
	Rank	4	3	2	7	1	9	8	6	5
F8	Best	8.0577E+02	8.0507E+02	8.0696E+02	8.3173E+02	8.0696E+02	8.2714E+02	8.4237E+02	8.0384E+02	8.0933E+02
	Average	8.2487E+02	8.1466E+02	8.2551E+02	8.4709E+02	8.2865E+02	8.5681E+02	8.8107E+02	8.1422E+02	8.3372E+02
	STD	1.0142E+01	6.5709E+00	1.0398E+01	8.4121E+00	1.2178E+01	1.1469E+01	1.0950E+01	9.0234E+00	1.5373E+01
	Rank	3	2	4	7	5	8	9	1	6
F9	Best	9.1527E+02	9.0797E+02	9.1194E+02	9.2856E+02	9.0895E+02	9.2023E+02	9.3255E+02	9.2172E+02	9.0802E+02
	Average	9.3154E+02	9.2103E+02	9.2680E+02	9.4997E+02	9.3387E+02	9.4587E+02	9.5971E+02	9.3257E+02	9.3254E+02
	STD	1.3263E+01	1.0719E+01	1.3313E+01	1.0481E+01	1.4888E+01	9.0763E+00	7.2704E+00	6.0098E+00	1.3190E+01
	Rank	2	1	3	8	6	7	9	5	4
F10	Best	1.0852E+03	1.1228E+03	1.0874E+03	1.6529E+03	1.1661E+03	1.3008E+03	1.7395E+03	1.1383E+03	1.1591E+03
	Average	1.4484E+03	1.4536E+03	1.5656E+03	2.1906E+03	1.6617E+03	1.6329E+03	2.1695E+03	1.4597E+03	1.5909E+03
	STD	2.4438E+02	2.9218E+02	2.0774E+02	2.1447E+02	2.7369E+02	2.0402E+02	1.9042E+02	1.8983E+02	2.2565E+02
	Rank	1	2	4	8	7	6	9	3	5
F11	Best	1.1958E+03	1.1228E+03	1.4510E+03	2.1970E+03	1.4591E+03	1.3177E+03	2.2910E+03	1.2425E+03	1.3875E+03
	Average	1.8978E+03	1.8918E+03	2.0549E+03	2.6162E+03	2.0594E+03	2.0484E+03	2.6121E+03	1.8371E+03	2.0763E+03
	STD	3.5080E+02	4.0378E+02	2.6432E-01	2.0701E+02	2.5991E+02	2.6187E+02	1.6891E+02	3.0596E+02	4.5980E+02
	Rank	3	2	5	9	6	4	8	1	7
F12	Best	1.2004E+03	1.2001E+03	1.2000E+03	1.2009E+03	1.2001E+03	1.2001E+03	1.2011E+03	1.2001E+03	1.2002E+03
	Average	1.2008E+03	1.2011E+03	1.2004E+03	1.2015E+03	1.2004E+03	1.2007E+03	1.2016E+03	1.2004E+03	1.2012E+03
	STD	3.5115E-01	7.6283E-01	3.0742E-01	3.0972E-01	3.0332E-01	3.7921E-01	3.4053E-01	1.6018E-01	6.3735E-01
	Rank	5	6	2	8	3	4	9	1	7
F13	Best	1.3003E+03	1.3002E+03	1.3002E+03	1.3005E+03	1.3001E+03	1.3005E+03	1.3017E+03	1.3002E+03	1.3002E+03
	Average	1.3005E+03	1.3003E+03	1.3004E+03	1.3008E+03	1.3004E+03	1.3032E+03	1.3035E+03	1.3005E+03	1.3004E+03
	STD	2.0830E-01	7.0961E-02	1.0997E-01	1.8979E-01	1.8024E-01	9.6517E-01	9.3114E-01	3.8420E-01	1.1445E-01
	Rank	5	1	3	7	4	8	9	6	2
F14	Best	1.4001E+03	1.4002E+03	1.4001E+03	1.4005E+03	1.4002E+03	1.4103E+03	1.4052E+03	1.4002E+03	1.4002E+03
	Average	1.4007E+03	1.4004E+03	1.4003E+03	1.4017E+03	1.4006E+03	1.4253E+03	1.4127E+03	1.4014E+03	1.4011E+03
	STD	9.7179E-01	2.1205E-01	1.5377E-01	9.0374E-01	3.3924E-01	8.0146E+00	5.2211E+00	2.6009E+00	2.8333E+00
	Rank	4	2	1	7	3	9	8	6	5
F15	Best	1.5017E+03	1.5006E+03	1.5006E+03	1.5064E+03	1.5006E+03	1.5327E+03	1.6277E+03	1.5025E+03	1.5019E+03
	Average	1.5075E+03	1.5027E+03	1.5018E+03	1.5175E+03	1.5024E+03	3.6063E+03	3.6496E+03	1.5179E+03	1.5162E+03
	STD	1.1348E+01	1.1222E+00	1.0137E+00	3.1130E+01	1.1303E+00	3.3031E+03	2.1669E+03	7.0145E+01	6.5191E+01
	Rank	4	3	1	6	2	8	9	7	5
F16	Best	1.6023E+03	1.6018E+03	1.6027E+03	1.6026E+03	1.6024E+03	1.6027E+03	1.6035E+03	1.6026E+03	1.6021E+03
	Average	1.6033E+03	1.6030E+03	1.6033E+03	1.6035E+03	1.6034E+03	1.6036E+03	1.6039E+03	1.6034E+03	1.6029E+03
	STD	5.0955E-01	4.5711E-01	2.9764E-01	3.0364E-01	4.2028E-01	3.8139E-01	1.2705E-01	2.7099E-01	4.1837E-01
	Rank	3	2	4	7	5	8	9	6	1
F17	Best	3.1014E+03	2.9363E+03	2.5608E+03	6.0987E+03	2.9885E+03	4.6446E+03	3.2649E+05	1.2756E+04	3.0619E+03
	Average	1.8104E+05	2.1061E+05	2.5376E+05	7.4956E+04	2.0833E+04	3.6084E+05	5.3652E+05	2.3358E+05	2.0340E+04
	STD	2.4750E+05	2.2606E+05	3.1246E+05	9.5037E+04	3.4033E+04	1.7417E+05	6.6016E+04	1.7199E+05	6.3606E+04
	Rank	4	5	7	3	2	8	9	6	1
F18	Best	2.0714E+03	2.0437E+03	2.0078E+03	8.5584E+03	1.9694E+03	1.9453E+03	1.4326E+04	5.8322E+03	1.9804E+03
	Average	7.9588E+03	1.1124E+04	1.2907E+04	5.8429E+04	1.2530E+04	1.3045E+04	7.1584E+05	1.1681E+04	1.2400E+04
	STD	6.5971E+03	8.1391E+03	1.0403E+04	7.8766E+04	8.6456E+03	8.8545E+03	1.7466E+06	2.9759E+03	7.9666E+03
	Rank	1	2	6	8	5	7	9	3	4
F19	Best	1.9020E+03	1.9016E+03	1.9017E+03	1.9038E+03	1.9023E+03	1.9057E+03	1.9098E+03	1.9021E+03	1.9019E+03
	Average	1.9037E+03	1.9033E+03	1.9046E+03	1.9064E+03	1.9041E+03	1.9302E+03	1.9227E+03	1.9039E+03	1.9039E+03

(continued on next page)

Table 14 (*continued*)

Fun	Measure	SCHO	GWO	ALO	SCA	SSA	AOA	RSA	SHO	GJO
	STD	1.3994E+00	1.3518E+00	1.6159E+00	1.3006E+00	8.3856E-01	2.2448E+01	1.3969E+01	1.1526E+00	1.4756E+00
	Rank	2	1	6	7	5	9	8	3	4
F20	Best	2.3633E+03	2.1451E+03	2.5264E+03	3.1752E+03	2.1380E+03	2.3057E+03	5.4105E+03	2.0701E+03	3.2904E+03
	Average	8.3031E+03	1.0084E+04	9.2225E+03	9.7014E+03	9.7167E+03	1.0159E+04	1.3561E+04	9.0505E+03	1.0303E+04
	STD	6.7945E+03	7.3670E+03	6.6161E+03	5.3456E+03	8.4594E+03	5.0908E+03	6.6068E+03	3.4371E+03	4.2416E+03
	Rank	1	6	3	4	5	7	9	2	8
F21	Best	2.6731E+03	3.9166E+03	2.6928E+03	3.6568E+03	2.9099E+03	3.7886E+03	8.1265E+03	2.4708E+03	3.2592E+03
	Average	1.0567E+04	1.0589E+04	1.2461E+04	1.4090E+04	1.1333E+04	1.1914E+06	4.8988E+05	1.5634E+04	1.1188E+04
	STD	6.8773E+03	5.4230E+03	8.0101E+03	7.6974E+03	8.2331E+03	2.6157E+06	7.8432E+05	2.6742E+04	5.2502E+03
	Rank	1	2	6	7	4	5	9	8	3
F22	Best	2.2268E+03	2.2269E+03	2.2258E+03	2.2324E+03	2.2261E+03	2.2300E+03	2.2874E+03	2.2241E+03	2.2377E+03
	Average	2.3255E+03	2.3114E+03	2.3100E+03	2.2730E+03	2.3392E+03	2.3879E+03	2.4178E+03	2.2873E+03	2.3448E+03
	STD	8.5554E+01	5.6703E+01	9.0055E+01	2.4781E+01	9.9059E+01	8.2830E+01	6.3444E+01	6.1918E+01	6.0807E+01
	Rank	5	4	3	1	6	8	9	2	7
F23	Best	2.3118E+03	2.6295E+03	2.6295E+03	2.6394E+03	2.6295E+03	2.5000E+03	2.5000E+03	2.5000E+03	2.5000E+03
	Average	2.4937E+03	2.6346E+03	2.6299E+03	2.6464E+03	2.6321E+03	2.5000E+03	2.5000E+03	2.6243E+03	2.6360E+03
	STD	3.4361E+01	5.7748E+00	2.0737E+00	4.9742E+00	4.7651E+00	0.0000E+00	0.0000E+00	4.2400E+01	2.6180E+01
	Rank	1	7	5	9	6	2	2	4	8
F24	Best	2.5243E+03	2.5104E+03	2.5189E+03	2.5355E+03	2.5157E+03	2.5531E+03	2.5855E+03	2.5227E+03	2.5171E+03
	Average	2.5601E+03	2.5525E+03	2.5661E+03	2.5602E+03	2.5413E+03	2.5906E+03	2.5995E+03	2.5720E+03	2.5494E+03
	STD	2.6101E+01	3.5929E+01	3.2619E+01	1.0368E+01	2.4651E+01	1.6631E+01	2.6493E+00	3.1375E+01	2.6034E+01
	Rank	4	3	6	5	1	8	9	7	2
F25	Best	2.6507E+03	2.6688E+03	2.6581E+03	2.6713E+03	2.6543E+03	2.6668E+03	2.7000E+03	2.6669E+03	2.6549E+03
	Average	2.6943E+03	2.6989E+03	2.6975E+03	2.6990E+03	2.6944E+03	2.6983E+03	2.7000E+03	2.6983E+03	2.6963E+03
	STD	1.0980E+01	7.4679E+00	1.1143E+01	9.2170E+00	1.2787E+01	6.1838E+00	0.0000E+00	6.7920E+00	1.1492E+01
	Rank	1	7	4	8	2	5	9	6	3
F26	Best	2.7002E+03	2.7001E+03	2.7001E+03	2.7004E+03	2.7001E+03	2.7021E+03	2.7023E+03	2.7001E+03	2.6589E+03
	Average	2.7005E+03	2.7035E+03	2.7036E+03	2.7008E+03	2.7004E+03	2.7187E+03	2.7067E+03	2.7037E+03	2.6990E+03
	STD	3.3159E-01	1.8226E+01	1.8247E+01	1.8025E-01	1.9347E-01	3.2584E+01	1.7633E+01	1.8188E+01	7.7868E+00
	Rank	3	5	6	4	2	9	8	7	1
F27	Best	2.7039E+03	2.7033E+03	2.7041E+03	2.7206E+03	2.7054E+03	2.9000E+03	2.9000E+03	2.7057E+03	2.7055E+03
	Average	2.8683E+03	3.0046E+03	3.0020E+03	3.0921E+03	3.0723E+03	2.9000E+03	3.0946E+03	3.0381E+03	3.0366E+03
	STD	7.2135E+01	1.3907E+02	1.6843E+02	7.2994E+01	1.0611E+02	0.0000E+00	1.5280E+02	1.6775E+02	1.5069E+02
	Rank	1	4	3	8	7	2	9	6	5
F28	Best	3.0000E+03	3.1693E+03	3.0000E+03	3.2288E+03	3.1689E+03	3.0000E+03	3.0000E+03	3.2552E+03	3.1700E+03
	Average	3.0000E+03	3.2515E+03	3.4337E+03	3.2989E+03	3.2463E+03	3.0000E+03	3.2883E+03	3.3653E+03	3.2901E+03
	STD	0.0000E+00	5.2321E+01	2.2707E+02	5.2046E+01	7.8945E+01	0.0000E+00	2.1808E+02	7.2176E+01	8.2737E+01
	Rank	1	4	9	7	3	1	5	8	6
F29	Best	3.1000E+03	3.2274E+03	3.1000E+03	4.0208E+03	3.3145E+03	3.1000E+03	3.1000E+03	3.1824E+03	3.1969E+03
	Average	3.9808E+03	3.5784E+05	1.2592E+05	1.8669E+04	1.3030E+05	5.4088E+05	3.6045E+05	6.8969E+05	4.5576E+03
	STD	1.9868E+03	8.2460E+05	6.6868E+05	1.5925E+04	4.7961E+05	1.2524E+06	1.0336E+06	1.6317E+06	1.4537E+03
	Rank	1	6	4	3	5	8	7	9	2
F30	Best	3.2000E+03	3.6011E+03	4.0699E+03	4.1293E+03	3.6506E+03	3.2000E+03	5.5967E+03	4.5365E+03	3.6537E+03
	Average	3.5583E+03	4.3455E+03	5.3679E+03	5.1091E+03	4.6713E+03	7.5115E+04	2.0846E+04	5.2521E+03	4.7573E+03
	STD	6.1821E+02	6.9083E+02	9.9491E+02	6.0757E+02	7.6307E+02	9.3183E+04	2.7452E+04	3.7912E+02	6.6832E+02
	Rank	1	2	7	5	3	9	8	6	4
Mean	Rank	2.73	3.73	4.17	6.43	3.80	6.87	8.13	4.93	4.33
Final	Ranking	1	2	4	7	3	8	9	6	5

between bearings (x_5), the diameter of the first shaft (x_6), and the diameter of the second shaft (x_7). The mathematical formulation of the speed reducer design problem is as follows.

Given the 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:

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

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

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

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

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

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

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

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

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

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

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

Variables range:

Table 15
Wilcoxon rank sum test between SCHO and the other algorithms for CEC 2014 test functions at $\alpha = 0.05$, $D = 10$.

F	GWO p-value	Sig.	ALO p-value	Sig.	SCA p-value	Sig.	SSA p-value	Sig.	AOA p-value	Sig.	RSA p-value	Sig.	SHO p-value	Sig.	GJO p-value	Sig.
F1	9.5E-03	+	2.2E-03	–	3.3E-01	~	3.8E-07	–	1.4E-07	+	4.5E-11	+	4.2E-01	~	9.1E-03	+
F2	8.5E-09	–	2.2E-03	–	1.5E-02	+	3.0E-11	–	4.1E-11	+	3.0E-11	+	1.4E-03	+	1.5E-01	~
F3	1.7E-02	+	2.2E-03	+	1.5E-02	+	3.6E-02	+	1.4E-04	+	9.5E-04	+	5.9E-04	–	7.6E-03	–
F4	4.8E-02	+	1.1E-03	+	1.7E-05	+	1.8E-04	+	3.0E-11	+	3.0E-11	+	2.1E-03	+	6.3E-01	~
F5	8.8E-01	~	2.2E-03	–	3.4E-02	+	4.5E-11	–	4.1E-11	–	3.0E-02	+	3.7E-11	–	4.2E-02	+
F6	6.7E-10	–	4.1E-02	+	9.6E-02	+	1.1E-01	~	7.8E-09	+	1.5E-09	+	2.0E-02	+	1.3E-04	–
F7	1.9E-07	–	2.2E-03	–	1.8E-03	+	3.0E-11	–	3.3E-11	+	1.1E-10	+	4.0E-01	~	4.0E-01	~
F8	7.8E-09	–	4.1E-02	+	1.7E-09	+	6.8E-02	~	2.2E-09	+	3.0E-11	+	1.3E-06	–	6.4E-01	~
F9	3.8E-10	–	2.6E-02	+	1.0E-04	+	1.1E-02	+	7.6E-03	+	1.2E-08	+	3.4E-02	+	3.1E-02	+
F10	1.1E-03	+	4.1E-02	+	4.2E-10	+	2.2E-02	+	1.2E-02	+	8.1E-10	+	5.9E-04	+	3.5E-02	+
F11	2.4E-04	–	5.1E-03	+	7.1E-08	+	1.5E-02	+	1.8E-01	~	1.1E-09	+	2.6E-02	–	1.3E-02	+
F12	4.4E-02	+	4.3E-03	–	1.2E-07	+	2.9E-09	–	3.2E-05	–	5.9E-06	+	1.4E-05	–	4.4E-02	+
F13	1.9E-09	–	4.8E-01	~	1.2E-06	+	2.2E-02	–	3.3E-11	+	3.0E-11	+	9.1E-03	+	3.8E-07	–
F14	4.9E-05	–	9.3E-02	–	1.8E-03	+	1.9E-01	~	1.3E-10	+	2.8E-08	+	4.5E-02	+	1.5E-01	~
F15	6.7E-10	–	2.2E-03	–	3.0E-02	+	6.1E-11	–	1.7E-09	+	2.4E-10	+	3.4E-02	+	4.0E-04	+
F16	4.1E-05	–	3.4E-02	+	7.0E-03	+	2.2E-03	+	7.0E-03	+	3.8E-07	+	4.7E-02	+	3.8E-02	–
F17	1.2E-05	+	3.1E-01	~	5.4E-02	~	2.0E-04	–	2.5E-02	+	6.5E-04	+	2.5E-01	~	2.9E-06	–
F18	6.6E-01	~	3.1E-01	~	1.6E-04	+	9.1E-01	~	2.4E-01	~	1.7E-06	+	9.9E-01	~	7.6E-01	~
F19	9.8E-05	–	9.4E-01	~	2.0E-07	+	3.0E-02	+	9.0E-11	+	3.0E-11	+	8.0E-03	+	9.1E-03	+
F20	2.0E-02	+	2.6E-03	+	4.8E-01	+	5.4E-02	+	2.4E-01	~	7.0E-03	+	4.4E-02	+	1.4E-03	+
F21	2.2E-02	+	2.3E-02	+	8.4E-01	~	6.2E-04	+	4.4E-05	+	7.1E-09	+	3.6E-04	+	1.9E-04	+
F22	1.6E-01	~	8.9E-02	~	1.2E-03	–	1.8E-02	+	2.0E-02	+	9.9E-03	+	9.0E-04	–	2.9E-03	+
F23	1.7E-12	+	2.2E-03	+	1.7E-12	+	1.7E-12	+	3.3E-01	~	3.3E-01	~	2.2E-11	+	6.2E-12	+
F24	6.7E-02	~	8.2E-01	~	2.2E-02	+	4.2E-06	–	1.6E-04	+	7.2E-07	+	3.0E-03	+	2.2E-01	~
F25	2.8E-03	+	1.6E-03	+	9.3E-07	+	4.2E-01	~	2.9E-02	+	4.8E-03	+	1.3E-01	~	2.4E-03	+
F26	3.8E-10	+	2.6E-02	+	6.2E-04	+	1.9E-05	–	2.6E-07	+	1.6E-07	+	2.8E-03	+	3.4E-05	–
F27	3.0E-07	+	1.8E-05	+	1.5E-07	+	2.1E-05	+	2.9E-01	~	1.2E-06	+	1.6E-08	+	1.9E-08	+
F28	1.2E-12	+	2.2E-03	+	1.2E-12	+	1.2E-12	+	NaN	~	1.7E-08	+	1.2E-12	+	1.2E-12	+
F29	2.0E-02	+	1.9E-02	+	9.2E-10	+	2.9E-02	+	8.3E-02	~	4.1E-10	+	5.5E-01	~	1.4E-02	+
F30	2.7E-05	+	5.3E-05	+	1.2E-07	+	3.1E-05	+	2.9E-11	+	1.7E-11	+	1.0E-09	+	2.7E-06	+
(W|L|T)	(15|11|4)		(17|7|6)		(26|1|3)		(15|10|5)		(21|2|7)		(29|0|1)		(18|6|6)		(17|6|7)	

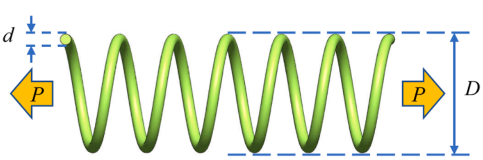


Fig. 10. Tension/compression spring design problem.

Table 16
Optimization results of SCHO and other algorithms for the tension/compression spring design problem.

Different Algorithms	Optimum variables x_1	x_2	x_3	Optimum weight ($f(x)$)
SCHO	0.0517422	0.3579972	11.2146238	0.0126656
SO [57]	0.0511	0.3418	12.2222	0.012672535
SCA [112]	0.137850353	1.26129434	12.53614501	0.0126763
WOA [54]	0.0512	0.3452	12.004	0.0126763
GSA [52]	0.050276	0.32368	13.52541	0.0127022
SSA [107]	0.051207	0.345215	12.004032	0.0126763
RO [113]	0.05137	0.349096	11.76279	0.012679
GJO [80]	0.0515793	0.354055	11.4484	0.01266752
PSO [114]	0.051728	0.357644	11.244543	0.0126747
AOA [115]	0.0508	0.3348	11.702	0.012681
SHO [109]	0.05194	0.36289	10.9358	0.01266644
DO [47]	0.051215	0.345416	11.983708	0.012669

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

The optimum weight of SCHO and the other 11 meta-heuristic optimization algorithms is shown in Table 19. The optimization results indicate that SCHO beats other algorithms and gets the optimum weight of 2995.2477, and the values of corresponding variables x_1, x_2, x_3, x_4, x_5, x_6, x_7 are 3.50008, 0.7, 17, 7.3, 7.72871, 3.35023, and 5.28736, respectively. Therefore, SCHO is still an effective optimization method for complex problems compared to other algorithms.

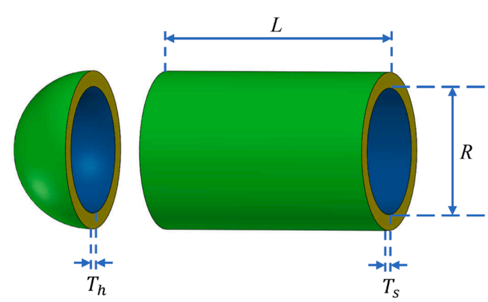


Fig. 11. Pressure vessel design problem.

3.3.5. Cantilever beam design problem

In this engineering design problem, SCHO is applied to minimize the optimum weight of the cantilever beam. This design problem, shown in Fig. 14, has five design variables. The mathematical formulation of the cantilever beam design problem is as follows.

Given the variables: x_1, x_2, x_3, x_4, x_5

Minimize:

$$f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$$

Subject to:

$$g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0,$$

Variables range:

$$0.01 \leq x_1,\ x_2,\ x_3,\ x_4,\ x_5 \leq 100$$

There is a constraint in this problem. The optimization results from

Table 17
Optimization results of SCHO and other algorithms for the pressure vessel design problem.

Different Algorithms	Optimum variables				Optimum cost ($f(x)$)
	x_1	x_2	x_3	x_4	
SCHO	0.7796836	0.3854124	40.39092	199.0132	5889.0061
RSA [108]	0.8400693	0.4189594	43.38117	161.5556	6034.7591
GWO [52]	0.8125	0.4345	42.0892	176.7587	6051.5639
WOA [54]	0.8125	0.4375	42.0982699	176.638998	6059.741
MVO [23]	0.8125	0.4375	42.0907382	176.73869	6060.8066
AO [105]	1.054	0.182806	59.6219	38.805	5949.2258
MPA [116]	0.8125	0.4375	42.098445	176.636607	6059.7144
SCA [62]	0.817577	0.417932	41.74939	183.5727	6137.3724
AOA [109]	0.8303737	0.4162057	42.75127	169.3454	6048.7844
HHO [117]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
CSA [118]	0.8125	0.4375	42.09844539	176.6365986	6059.714363
ACO [119]	0.8125	0.4375	42.103624	176.572656	6059.0888

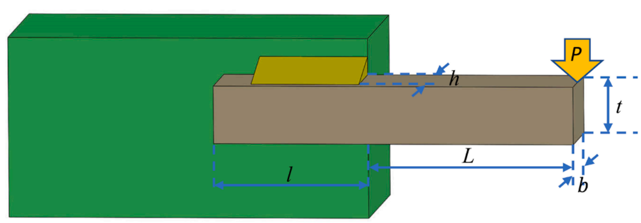


Fig. 12. Welded beam design problem.

Table 18
Optimization results of SCHO and other algorithms for the welded beam design problem.

Different Algorithms	Optimum variables				Optimum cost ($f(x)$)
	x_1	x_2	x_3	x_4	
SCHO	0.20565	3.47312	9.03685	0.20573	1.72516
GWO [52]	0.205676	3.478377	9.03681	0.205778	1.72624
WOA [54]	0.205396	3.484293	9.037426	0.206276	1.730499
HHO [117]	0.204039	3.531061	9.027463	0.206147	1.731991
SCA [62]	0.204695	3.536291	9.00429	0.210025	1.759173
CPSO [114]	0.202369	3.544214	9.04821	0.205723	1.72802
GSA [80]	0.182129	3.856979	10	0.202376	1.87995
MVO [23]	0.205463	3.473193	9.044502	0.205695	1.72645
HGSO [120]	0.2054	3.4476	9.0269	0.206	1.726
GA [62]	0.164171	4.032541	10	0.223647	1.873971
GJO [80]	0.20562	3.4719	9.0392	0.20572	1.72522
SHO [109]	0.20585	3.46946	9.03276	0.20591	1.7259

SCHO and other reported meta-heuristic algorithms are shown in Table 20. It can be observed that SCHO gets the optimum weight of 1.3033, and the corresponding five variables x_1, x_2, x_3, x_4, x_5 are 5.9763, 4.8878, 4.4573, 3.4732, and 2.1447, respectively, while the optimum weight of other comparison algorithms is above 1.33, indicating that SCHO is significantly superior to other algorithms.

3.3.6. *Three-bar truss design problem*

Three-bar truss is a classical problem in civil engineering, in which two variables are selected to minimize the weight of truss structure subjected to deflection, stress, and buckling constraints. The geometric figure of three-bar truss is shown in Fig. 15, and SCHO is applied to optimize this problem by adjusting two variables including $A_1(x_1)$ and $A_2(x_1)$. The mathematical formulation of the three-bar truss design problem is as follows.

Given the variables: x_1, x_2

Minimize:

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

Subject to:

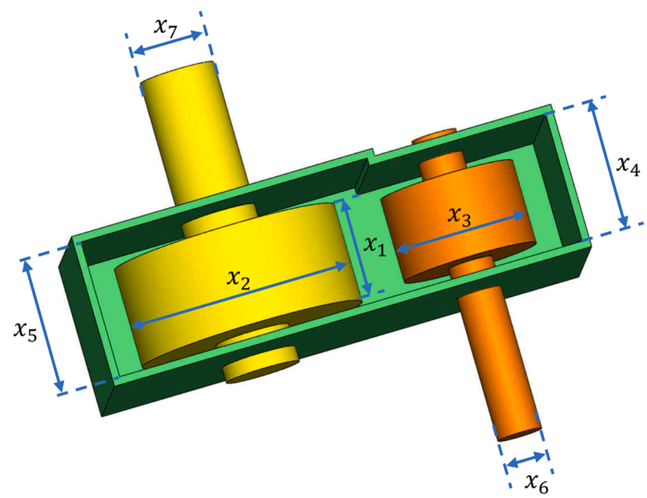


Fig. 13. Speed reducer design problem.

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

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

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

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

Variables range:

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

The optimization results of SCHO and other algorithms are shown in Table 21. It can be observed that SCHO is competitive compared with other comparison algorithms, since SCHO obtains the optimum weight of 263.8958476 when variables of x_1, x_2 take values of 0.78866420 and 0.40827926, respectively.

3.4. *Summary of results*

The analysis results of four subordinate models show that SCHO with four subordinate models can obtain the best optimization ability. It is observed from the qualitative analysis that SCHO presents strong exploration ability in the early iterations, good exploitation ability in the later iterations, and fast convergence speed. Compared with eight meta-heuristic algorithms using unimodal and multi-modal test functions (F1-F13), the exploitation and exploration of SCHO are superior to other

Table 19
Optimization results of SCHO and other algorithms for the speed reducer design problem.

Different Algorithms	Optimum variables							Optimum weight ($f(x)$)
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
SCHO	3.50008	0.7	17	7.3	7.72871	3.35023	5.28736	2995.2477
MFPA [121]	3.5	0.7	17	7.30812	7.8005	3.35021	5.28668	2996.2195
PSO [122]	3.5001	0.7	17.0002	7.5177	7.7832	3.3508	5.2867	3145.922
SCA [62]	3.508755	0.7	17	8.3	3.46102	5.289213	5.300734	3030.563
GWO [52]	3.501	0.7	17	7.3	7.811013	3.350704	5.287411	2997.81965
AO [55]	3.5021	0.7	17	7.3099	7.7476	3.3641	5.2994	3007.7328
HGSO [120]	3.498	0.71	17.02	7.67	7.81	3.36	5.289	2997.1
MDE [123]	3.50001	0.7	17	7.300156	7.800027	3.350221	5.286685	2996.35669
AOA [105]	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157
POA [124]	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.3482
HHO [57]	3.4981	0.7	17	7.6398	7.8	3.3582	5.2853	3000.67208
LGSI4 [125]	3.501	0.7	17	7.3	7.8	3.350214	5.286683	2996.348205

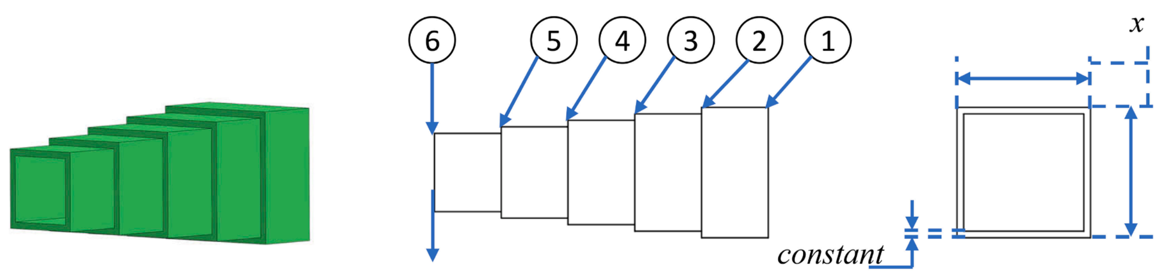


Fig. 14. Cantilever beam design problem.

Table 20
Optimization results of SCHO and other algorithms for the cantilever beam design problem.

Different Algorithms	Optimum variables					Optimum weight ($f(x)$)
	x_1	x_2	x_3	x_4	x_5	
SCHO	5.9763	4.8878	4.4573	3.4732	2.1447	1.3033
RSA [108]	6.0231	5.4457	4.277	3.5853	2.1767	1.3386
SOS [126]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
CS [51]	6.0089	5.3049	4.5023	3.5077	2.1504	1.3399
ALO [106]	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
MFO [53]	5.984871773	5.316726924	4.497332586	3.513616468	2.161620293	1.339988086
MMA [127]	6.01	5.3	4.49	3.49	2.15	1.34
SHO [109]	6.0049	5.3227	4.4737	3.5065	2.16637	1.339987
GCA_I [127]	6.01	5.304	4.49	3.498	2.15	1.34
GCA_II [127]	6.01	5.3	4.49	3.49	2.15	1.34
SSA [107]	6.015134526	5.309304676	4.495006716	3.501426286	2.152787908	1.339956391

algorithms. As a results, SCHO is ranked first in the 23 classical benchmark functions. The results of convergence analysis indicate that SCHO exhibits fast convergence speed to obtain the optimal solution on most functions, which are benefit by the first and second phase of exploration and exploitation, bounded strategy, and good switch mechanism. The results of the 13 benchmark functions at high dimensions show that SCHO still maintains its superiority, and even exhibits more stable compared with other algorithms. In addition, SCHO performs better than other algorithms for the CEC 2014 test functions, especially F23-F30, which indicates SCHO still keeps its superiority for solving complex problems. It is also concluded from six engineering design problems that SCHO is very effective for optimizing these kind of problems. Hence, SCHO outperforms the other algorithms for the 23 classical benchmark functions, CEC 2014, and six engineering design problems, and shows a good balance ability between exploration and exploitation to obtain optimal solutions.

4. Conclusion and future works

This study proposes a novel meta-heuristic optimization algorithm, named Sinh Cosh Optimizer (SCHO), according to the mathematical inspiration of the characteristics of Sinh and Cosh. The SCHO mathematical model comprises four parts: the first and second phases of exploration and exploitation, the bounded search strategy, and the switching mechanism. The Sinh and Cosh are integrated into these parts to model the SCHO algorithm.

The optimization performance of SCHO to look for the optimal solution was validated for the 23 classical benchmark functions. The analysis results of four subordinate models show that SCHO with four models can exhibit the best performance. The qualitative analysis of trajectory of first search agent on benchmark functions showed that SCHO exhibited good exploration in the early iterations and strong exploitation in the later iterations, while the search history, average fitness and convergence curve verified the fast convergence of SCHO. The optimization results from SCHO were compared with eight meta-heuristic algorithms like GWO, ALO, SCA, SSA, AOA, RSA, SHO, and GJO. The mean ranking and the number of optimal solutions all are ranked first, indicating that SCHO possesses a great performance in looking for the optimal solution. The results of the unimodal tested functions demonstrated the superiority of SCHO in exploitation compared to other algorithms. At the same time, SCHO outperformed other algorithms in term of exploration of the multi-modal tested

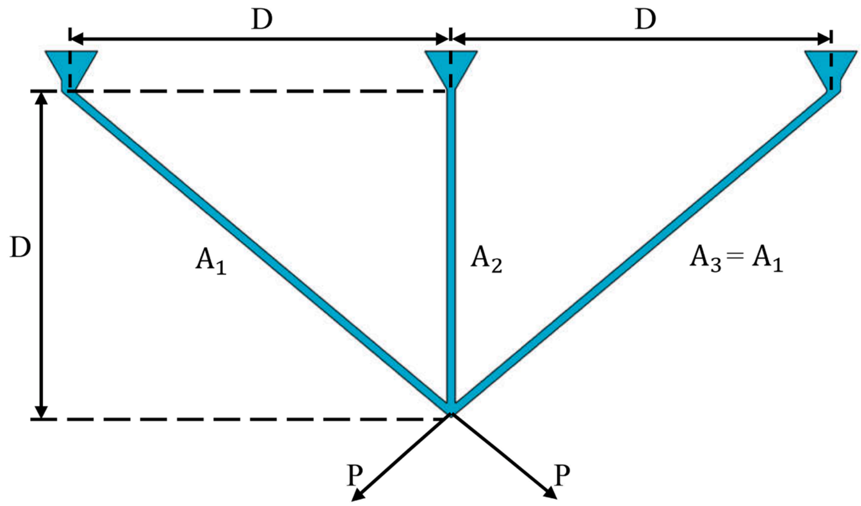


Fig. 15. Three-bar truss design problem.

Table 21
Optimization results of SCHO and other algorithms for the three-bar truss design problem.

Different Algorithms	Optimum variables		Optimum weight ($f(x)$)
	x_1	x_2	
SCHO	0.78866420	0.40827926	**263.8958476**
AOA [105]	0.79369	0.39426	263.9154
CS [51]	0.78867	0.40902	263.97156
MBA [128]	0.7885650	0.4085597	263.8958522
MVO [23]	0.78860276	0.40845307	263.8958499
MFO [53]	0.788244770931922	0.409466905784741	263.895979682
WOA [73]	0.789050544	0.407187512	263.8959474
GOA [129]	0.788897555578973	0.407619570115153	263.895881496069
Ray and Sain [130]	0.795	0.395	264.3
SCA [131]	0.78669	0.41426	263.9348
DA [67]	0.7883714	0.409108	263.89591

functions. Hence, the exploration and exploitation are consistent with qualitative analysis, and the good results verify the effectiveness of the proposed switching mechanism. Moreover, the fast convergence ability of SCHO is also confirmed by the convergence curves. It is concluded that the three search strategies and good switching mechanism ensure a satisfactory convergence behavior and good optima avoidance to find the optimal solutions. To evaluate the scalability of the algorithm, 13 classical benchmark functions with 100 and 500 dimensions are optimized by SCHO and comparative algorithms. The results show that SCHO can retain the superiority of low dimension compared with other algorithms, and dimensional changes have less impact on SCHO. In addition, the results of SCHO on CEC 2014 test function indicate its superiority for solving complex problems.

In order to determine the efficiency of SCHO, six classical engineering design problems with inequal constraints were selected, including tension/compression spring design problem, pressure vessel design problem, welded beam design problem, speed reducer design problem, cantilever beam design problem and three-bar truss design problem. The results indicated that SCHO possessed a better ability to minimize the optimum weight or cost than other meta-heuristic algorithms, which indicates that SCHO still performs well in terms of these problems with unknown, challenging search spaces. Therefore, SCHO is a potential optimization algorithm for solving the constrained and unconstrained problems.

Overall, SCHO is a competitive algorithm with good balance between exploration and exploitation to escape from local optimum and find the optimal solution. However, SCHO is applied for solving the simple-objective problems, while its performance on multi-objective problems is unknown. Also, it is undeniable that a good balance strategy between exploration and exploitation remains an intractable challenge in meta-heuristics algorithms. Therefore, for future research, we will continue to apply it to multi-objective problems, and further explore the strategy how to better balance exploration and exploitation.

CRediT authorship contribution statement

Jianfu Bai: Investigation, Methodology, Validation, Writing – original draft. **Yifei Li:** Formal analysis. **Mingpo Zheng:** Visualization. **Samir Khatir:** Writing – review & editing. **Brahim Benaissa:** Writing – review & editing. **Laith Abualigah:** Writing – review & editing. **Magd Abdel Wahab:** Supervision, Conceptualization, Validation, Writing – review & editing.

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.

Acknowledgement

The authors would like to acknowledge the financial support of the grants from the China Scholarship Council (202108440192).

References

[1] R.M. Rizk-Allah, O. Saleh, E.A. Hagag, A.A.A. Mousa, Enhanced tunicate swarm algorithm for solving large-scale nonlinear optimization problems, Int. J. Comput. Intell. Syst. 14 (1) (2021).
[2] J. Jiang, Z. Zhao, Y. Liu, W. Li, H. Wang, DSGWO: an improved grey wolf optimizer with diversity enhanced strategy based on group-stage competition and balance mechanisms, Knowl. Based Syst. 250 (2022).
[3] R. Wehrens, L.M. Buydens, Classical and nonclassical optimization methods, Encyclopedia Anal. Chem. (2000) 9678–9689.
[4] L. YiFei, C. MaoSen, H.T. Ngoc, S. Khatir, M. Abdel Wahab, Multi-parameter identification of concrete dam using polynomial chaos expansion and slime mould algorithm, Comput. Struct. 281 (2023).
[5] M. Khishe, M.R. Mosavi, Chimp optimization algorithm, Expert Syst. Appl. 149 (2020).
[6] H.L. Minh, T. Sang-To, G. Theraulaz, M. Abdel Wahab, T. Cuong-Le, Termite life cycle optimizer, Expert Syst. Appl. 213 (2023).
[7] N.S. Jaddi, S. Abdullah, Global search in single-solution-based metaheuristics, Data Technol. Appl. 54 (3) (2020) 275–296.
[8] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inf. Sci. 237 (2013) 82–117.
[9] B. Doğan, T. Ölmez, A new metaheuristic for numerical function optimization: vortex Search algorithm, Inf. Sci. 293 (2015) 125–145.
[10] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, The Social Engineering Optimizer (SEO), Eng. Appl. Artif. Intell. 72 (2018) 267–293.
[11] N.H. Abdul Aziz, Z. Ibrahim, N.A. Ab Aziz, M.S. Mohamad, J. Watada, Single-solution simulated Kalman filter algorithm for global optimisation problems, Sādhanā 43 (2018) 1–15.
[12] R. Rajakumar, P. Dhavachelvan, T. Vengattaraman, A survey on nature inspired meta-heuristic algorithms with its domain specifications, in: Proceedings of the 2016 international conference on communication and electronics systems (ICCES), IEEE, 2016, pp. 1–6.
[13] L. YiFei, et al., Structure damage identification in dams using sparse polynomial chaos expansion combined with hybrid K-means clustering optimizer and genetic algorithm, Eng. Struct. 283 (2023).
[14] K. De Jong, Learning with genetic algorithms: an overview, Mach. Learn. 3 (1988) 121–138.
[15] J.R. Koza, Genetic Programming, On the Programming of Computers by Means of Natural Selection. A Bradford Book, MIT Press, 1992.
[16] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341.
[17] K. Juste, H. Kita, E. Tanaka, J. Hasegawa, An evolutionary programming solution to the unit commitment problem, IEEE Trans. Power Syst. 14 (4) (1999) 1452–1459.
[18] H.G. Beyer, H.P. Schwefel, Evolution strategies–a comprehensive introduction, Nat. Comput. 1 (2002) 3–52.
[19] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[20] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[21] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[22] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3–4) (2010) 267–289.
[23] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, Neural. Comput. Appl. 27 (2) (2015) 495–513.
[24] Z. Wei, C. Huang, X. Wang, T. Han, Y. Li, Nuclear reaction optimization: a novel and powerful physics-based algorithm for global optimization, IEEE Access 7 (2019) 66084–66109.
[25] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowl. Based Syst. 163 (2019) 283–304.
[26] J.L.J. Pereira, M.B. Francisco, C.A. Diniz, G. Antônio Oliver, S.S. Cunha, G. F. Gomes, Lichtenberg algorithm: a novel hybrid physics-based meta-heuristic for global optimization, Expert Syst. Appl. 170 (2021).
[27] H. Karami, M.V. Anaraki, S. Farzin, S. Mirjalili, Flow Direction Algorithm (FDA): a novel optimization approach for solving optimization problems, Comput. Ind. Eng. 156 (2021).
[28] M. Abdel-Basset, R. Mohamed, K.M. Sallam, R.K. Chakrabortty, Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm, Mathematics 10 (19) (2022).
[29] F.A. Hashim, R.R. Mostafa, A.G. Hussien, S. Mirjalili, K.M. Sallam, Fick's Law Algorithm: a physical law-based algorithm for numerical optimization, Knowl. Based Syst. 260 (2023).
[30] M. Abdel-Basset, R. Mohamed, S.A.A. Azeem, M. Jameel, M. Abouhawwash, Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion, Knowl. Based Syst. 268 (2023).
[31] L. Deng, S. Liu, Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design, Expert Syst. Appl. 225 (2023).
[32] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[33] Y. Xu, Y. Peng, X. Su, Z. Yang, C. Ding, X. Yang, Improving teaching–learning-based-optimization algorithm by a distance-fitness learning strategy, Knowl. Based Syst. 257 (2022).
[34] S. Kumar, G.G. Tejani, N. Pholdee, S. Bureerat, P. Jangir, Multi-Objective teaching-learning-based optimization for structure optimization, Smart Sci. 10 (1) (2021) 56–67.
[35] A.H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, in: Presented at the Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition, 2009.
[36] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, Comput. Struct. 139 (2014) 18–27.
[37] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: a new heuristic optimization method, Neurocomputing 221 (2017) 123–137.
[38] S.H. Samareh Moosavi, V.K. Bardsiri, Poor and rich optimization algorithm: a new human-based and multi populations algorithm, Eng. Appl. Artif. Intell. 86 (2019) 165–181.
[39] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern. 11 (7) (2019) 1501–1529.
[40] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowl. Based Syst. 191 (2020).
[41] A. Naik, S.C. Satapathy, Past present future: a new human-based algorithm for stochastic optimization, Soft. Comput. 25 (20) (2021) 12915–12976.
[42] E. Trojovska, M. Dehghani, A new human-based metahurestic optimization method based on mimicking cooking training, Sci. Rep. 12 (1) (Sep 1 2022) 14861.
[43] I. Faridmehr, M.L. Nehdi, I.F. Davoudkhani, A. Poolad, Mountaineering team-based optimization: a novel human-based metaheuristic algorithm, Mathematics 11 (5) (2023).
[44] M. Verij kazemi, E. Fazeli Veysari, A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm, Expert Syst. Appl. 193 (2022).
[45] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (4) (2018) 2191–2233.
[46] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021).
[47] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion Optimizer: a nature-inspired metaheuristic algorithm for engineering applications, Eng. Appl. Artif. Intell. 114 (2022).
[48] A.R. Kashani, R. Chiong, S. Mirjalili, A.H. Gandomi, Particle swarm optimization variants for solving geotechnical problems: review and comparative analysis, Arch. Comput. Meth. Eng. 28 (3) (2020) 1871–1927.
[49] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization-artificial ants as a computational intelligence technique, IEEE Comput. Intell. Mag. (2006).
[50] X.S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Comput. 29 (5) (2012) 464–483.
[51] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2011) 17–35.
[52] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[53] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl. Based Syst. 89 (2015) 228–249.
[54] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67.
[55] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021).
[56] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 114 (2022).
[57] F.A. Hashim, A.G. Hussien, Snake Optimizer: a novel meta-heuristic optimization algorithm, Knowl. Based Syst. 242 (2022).
[58] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, Knowl. Based Syst. 262 (2023).
[59] T. Sang-To, H. Le-Minh, M. Abdel Wahab, C.L. Thanh, A new metaheuristic algorithm: shrimp and Goby association search algorithm and its application for damage identification in large-scale and complex structures, Adv. Eng. Software 176 (2023).
[60] X.S. Yang, Flower pollination algorithm for global optimization, in: Proceedings of the Unconventional Computation and Natural Computation: 11th International Conference, UCNC 2012, Orléan, France, September 3-7, 2012. Proceedings 11, Springer, 2012, pp. 240–249.
[61] X.B. Meng, X.Z. Gao, L. Lu, Y. Liu, H. Zhang, A new bio-inspired optimisation algorithm: bird Swarm Algorithm, J. Exp. Theor. Artif. Intell. 28 (4) (2016) 673–687.
[62] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Software 114 (2017) 48–70.
[63] G.F. Gomes, S.S. da Cunha, A.C. Ancelotti, A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates, Eng. Comput. 35 (2) (2018) 619–626.
[64] A. Cheraghalipour, M. Hajiaghaei-Keshteli, M.M. Paydar, Tree Growth Algorithm (TGA): a novel approach for solving optimization problems, Eng. Appl. Artif. Intell. 72 (2018) 393–414.

[65] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[66] V. Hayyolalam, A.A. Pourhaji Kazem, Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020).
[67] A. Mohammadi-Balani, M. Dehghan Nayeri, A. Azar, M. Taghizadeh-Yazdi, Golden eagle optimizer: a nature-inspired metaheuristic algorithm, Comput. Ind. Eng. 152 (2021).
[68] M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems, Knowl. Based Syst. 243 (2022).
[69] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, Comput. Methods Appl. Mech. Eng. 391 (2022).
[70] A.H. Rabie, A.I. Saleh, N.A. Mansour, Red piranha optimization (RPO): a natural inspired meta-heuristic algorithm for solving complex optimization problems, J. Ambient Intell. Humaniz. Comput. 14 (6) (2023) 7621–7648.
[71] M. Azizi, S. Talatahari, A.H. Gandomi, Fire hawk optimizer: a novel metaheuristic algorithm, Artif. Intell. Rev. 56 (1) (2022) 287–363.
[72] I. Naruei, F. Keynia, Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems, Eng. Comput. 38 (S4) (2021) 3025–3056.
[73] H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, Appl. Math. Model. 71 (2019) 45–59.
[74] B. Benaissa, N.A. Hocine, S. Khatir, M.K. Riahi, S. Mirjalili, YUKI algorithm and POD-RBF for elastostatic and dynamic crack identification, J. Comput. Sci. (2021), 101451.
[75] N. Amoura, B. Benaissa, M. Al Ali, S. Khatir, Deep neural network and YUKI algorithm for inner damage characterization based on elastic boundary displacement, in: Proceedings of the International Conference of Steel and Composite for Engineering Structures: ICSCES 2022, Springer, 2023, pp. 220–233.
[76] M.I. Shirazi, S. Khatir, B. Benaissa, S. Mirjalili, M.A. Wahab, Damage assessment in laminated composite plates using modal strain energy and YUKI-ANN algorithm, Compos. Struct. (2022), 116272.
[77] A. Khatir, et al., A new hybrid PSO-YUKI for double crack identification in CFRP cantilever beam, Compos. Struct. (2023), 116803.
[78] H. Parmaksiz, U. Yuzgec, E. Dokur, N. Erdogan, Mutation based improved dragonfly optimization algorithm for a neuro-fuzzy system in short term wind speed forecasting, Knowl. Based Syst. 268 (2023).
[79] J. Adhikary, S. Acharyya, Randomized Balanced Grey Wolf Optimizer (RBGWO) for solving real life optimization problems, tAppl. Soft. Comput.. 117 (2022).
[80] N. Chopra, M. Mohsin Ansari, Golden jackal optimization: a novel nature-inspired optimizer for engineering applications, Expert Syst. Appl. 198 (2022).
[81] A. Ghasemi-Marzbali, A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm, Soft. Comput. 24 (17) (2020) 13003–13035.
[82] H.L. Minh, T. Sang-To, M. Abdel Wahab, T. Cuong-Le, A new metaheuristic optimization based on K-means clustering algorithm and its application to structural damage identification, Knowl. Based Syst. 251 (2022).
[83] M. Oszust, Enhanced marine predators algorithm with local escaping operator for global optimization, Knowl. Based Syst. 232 (2021).
[84] S. Chakraborty, A.K. Saha, R. Chakraborty, M. Saha, An enhanced whale optimization algorithm for large scale optimization problems, Knowl. Based Syst. 233 (2021).
[85] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, Y. Deng, An Improved Moth-Flame Optimization algorithm with hybrid search phase, Knowl. Based Syst. 191 (2020).
[86] Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, Y. Liu, A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems, Mathematics 10 (9) (2022).
[87] S.K. Joshi, Levy flight incorporated hybrid learning model for gravitational search algorithm, Knowl. Based Syst. 265 (2023).
[88] G. Dhiman, SSC: a hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications, Knowl. Based Syst. 222 (2021).
[89] G. Hu, B. Du, X. Wang, G. Wei, An enhanced black widow optimization algorithm for feature selection, Knowl. Based Syst. 235 (2022).
[90] R.R. Mostafa, M.A. Gaheen, M. Abd ElAziz, M.A. Al-Betar, A.A. Ewees, An improved gorilla troops optimizer for global optimization problems and feature selection, Knowl. Based Syst. 269 (2023).
[91] L. Abualigah, et al., Efficient text document clustering approach using multi-search arithmetic optimization algorithm, Knowl. Based Syst. 248 (2022).
[92] Y. Zhou, H. Wu, Q. Luo, M. Abdel-Baset, Automatic data clustering using nature-inspired symbiotic organism search algorithm, Knowl. Based Syst. 163 (2019) 546–557.
[93] E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy cuckoo optimization algorithm, Appl. Soft. Comput.. 41 (2016) 15–21.
[94] E.H. Houssein, et al., An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation, Knowl. Based Syst. 229 (2021).
[95] D. Yousri, M.A. Elaziz, S. Mirjalili, Fractional-order calcul Keshteli us-based flower pollination algorithm with local search for global optimization and image segmentation, Knowl. Based Syst. 197 (2020).
[96] Z. Yang, K. Li, Y. Guo, H. Ma, M. Zheng, Compact real-valued teaching-learning based optimization with the applications to neural network training, Knowl. Based Syst. 159 (2018) 51–62.
[97] V.K. Bohat, K.V. Arya, An effective best-guided gravitational search algorithm for real-parameter optimization and its application in training of feedforward neural networks, Knowl. Based Syst. 143 (2018) 192–207.
[98] P. Singh, R. Kottath, G.G. Tejani, Ameliorated follow the leader: algorithm and application to truss design problem, Structures 42 (2022) 181–204.
[99] C. Qu, W. Gai, J. Zhang, M. Zhong, A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning, Knowl. Based Syst. 194 (2020).
[100] M. Reda, A. Onsy, M.A. Elhosseini, A.Y. Haikal, M. Badawy, A discrete variant of cuckoo search algorithm to solve the Travelling Salesman Problem and path planning for autonomous trolley inside warehouse, Knowl. Based Syst. 252 (2022).
[101] S.K.R. Kanna, K. Sivakumar, N. Lingaraj, Development of deer hunting linked earthworm optimization algorithm for solving large scale traveling salesman problem, Knowl. Based Syst. 227 (2021).
[102] S. Faramarzi-Oghani, P. Dolati Neghabadi, E.G. Talbi, R. Tavakkoli-Moghaddam, Meta-heuristics for sustainable supply chain management: a review, Int. J. Prod. Res. (2022) 1–31.
[103] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[104] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl. Based Syst. 96 (2016) 120–133.
[105] L. Abualigah, A. Diabat, S. Mirjalili, M.Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Eng. 376 (2021).
[106] S. Mirjalili, The ant lion optimizer, Adv. Eng. Software 83 (2015) 80–98.
[107] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Software 114 (2017) 163–191.
[108] L. Abualigah, M.A. Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022).
[109] S. Zhao, T. Zhang, S. Ma, M. Wang, Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems, Appl. Intell. (2022).
[110] M.W. Li, J. Geng, W.C. Hong, L.D. Zhang, Periodogram estimation based on LSSVR-CCPSO compensation for forecasting ship motion, Nonlinear Dyn. 97 (4) (2019) 2579–2594.
[111] F.Y. Arini, S. Chiewchanwattana, C. Soomlek, K. Sunat, Joint Opposite Selection (JOS): a premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single-objective problems, Expert Syst. Appl. 188 (2022).
[112] A.E. Ezugwu, J.O. Agushaka, L. Abualigah, S. Mirjalili, A.H. Gandomi, Prairie dog optimization algorithm, Neural. Comput. Appl. 34 (22) (2022) 20017–20065.
[113] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[114] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[115] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (3) (2020) 1531–1551.
[116] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020), 113377.
[117] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[118] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[119] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. (2010).
[120] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: a novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[121] O.K. Meng, O. Pauline, S.C. Kiong, H.A. Wahab, N. Jafferi, Application of modified flower pollination algorithm on mechanical engineering design problem, in: Proceedings of the IOP conference series: materials science and engineering 165, IOP Publishing, 2017, 012032.
[122] S. Stephen, D. Christu, A. Dalvi, Design optimization of weight of speed reducer problem through matlab and simulation using ansys, Int. J. Mech. Eng. Technol. 9 (2018) 339–349.
[123] V.K. Kamboj, A. Nandi, A. Bhadoria, S. Sehgal, An intensify Harris Hawks optimizer for numerical and engineering optimization problems, Appl. Soft. Comput.. 89 (2020).
[124] P. Trojovsky, M. Dehghani, Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications, Sensors (Basel) 22 (3) (Jan 23 2022).
[125] A. Baykasoğlu, Ş. Akpinar, Weighted Superposition Attraction (WSA): a swarm intelligence algorithm for optimization problems – Part 1: unconstrained optimization, tAppl. Soft. Comput.. 56 (2017) 520–540.
[126] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[127] H. Chickermane, H.C. Gea, Structural optimization using a new local approximation method, Int. J. Numer. Methods Eng. 39 (5) (1996) 829–846.
[128] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems, Appl. Soft. Comput.. 13 (5) (2013) 2592–2612.

[129] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Software 105 (2017) 30–47.
[130] T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, Eng. Optim. 33 (6) (2001) 735–748, 2001/08/01.
[131] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2019) 210–230.