IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 September 2023, accepted 23 October 2023, date of publication 26 October 2023, date of current version 6 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327732

RESEARCH ARTICLE

IMSCSO: An Intensified Sand Cat Swarm Optimization With Multi-Strategy for Solving Global and Engineering Optimization Problems

XUEWEI LI[1,2], YONGLAN QI[1], QIAN XING[1], AND YONGTAO HU[3]

[1]School of Intelligent Engineering, Henan Institute of Technology, Xinxiang, Henan 453003, China
[2]Weihua Group Company Ltd., Changyuan 453400, China
[3]School of Electrical Engineering and Automation, Henan Institute of Technology, Xinxiang, Henan 453003, China

Corresponding author: Xuewei Li (lixuewei@hait.edu.cn)

This work was supported in part by the Science and Technology Key Project in Xinxiang City under Grant GG2021017, and in part by the Doctoral Research Start-Up Fund of Henan Institute of Technology under Grant 9001/KQ1908.

ABSTRACT Optimization challenges are becoming more complex as the world advances. Since deterministic and heuristic approaches are no longer sufficient to deal with such complex problems, metaheuristics have recently emerged as a viable option to address optimization difficulties. Since Sand Cat Swarm Optimization (SCSO) is a famous meta-heuristic algorithm, SCSO has a weak ability to balance search between exploration and exploitation and slow convergence, so it may not be effective in finding the global optima, particularly for complex problems. Hence, this paper proposes an intensified SCSO with multiple strategies (IMSCSO). The performance of the IMSCSO algorithm was evaluated on 23 standard test functions and test suites of CEC 2017, CEC 2019, and CEC 2020. Experimental results show that the IMSCSO algorithm performs significantly better than or is on par with other state-of-the-art optimizers. The statistical results obtained from the Wilcoxon signed-rank test and the Friedman test also indicate that the IMSCSO algorithm has a high ability to significantly outperform and rank first among all methods. Moreover, seven typical engineering issues were employed to estimate the efficacy of IMSCSO in optimizing constrained problems. The experimental findings show that the suggested IMSCSO method can efficiently handle real-world application issues.

INDEX TERMS Sand cat swarm optimization, hybrid opposition-based learning, joint opposite selection, benchmark functions.

I. INTRODUCTION

In real life, several optimization issues have evolved in numerous industries and scientific and technological disciplines, such as asset allocation [1], batch processing machines [2], photovoltaic power prediction [3], and tourism trips [4]. These optimization challenges get increasingly difficult and diverse as humans and industries progress. The greater and more intricate the problem, the more difficult it is to tackle. As a result, academics are eager to create better optimization approaches for handling these difficult issues. At the moment, deterministic and stochastic approximation approaches are useful for solving complicated issues. Deterministic algorithms [5] use specified mathematical functions to provide the same solution for different inputs to a given issue. For example, gradient descent method [6], Quasi-Newton method [7], Levenberg-Marquardt method [8], etc. Although the deterministic techniques ensure the optimal solution of the optimization issue, they have the premature convergence and are prone to falling into the local traps, particularly for high-dimensional and large-scale multimodal situations. The metaheuristic algorithm, as one of the most common

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

branches of approximation algorithms, is not guaranteed to discover the optimal solution, but the produced answer may be closer to the ideal solution. It has the advantages of simplicity, efficiency, and low complexity, and it can solve the flaws of deterministic approaches such as premature convergence and local stagnation by using random operators. Many metaheuristic algorithms are utilized to tackle challenging issues due to their superior properties. On the other hand, the No Free Lunch (NFL) theorems [9] show that no specific metaheuristic approach is capable of providing the optimum solution to every optimization problem. As a result, it is still essential to focus on developing sophisticated metaheuristic algorithms to address various optimization challenges.

There are many kinds of classification methods for metaheuristic algorithms, but there is no standard and general classification method [10]. Generally, metaphor-inspired approaches divide them into these categories: evolutionary-based algorithms (EBAs), swarm-based algorithms (SBAs), human-based algorithms (HBAs), physics-based algorithms (PBAs), math-based algorithms (MBAs), and game-based algorithms (GBAs).

EBAs are essentially population-based methods that use the population's interaction to seek out the global optimal solution in the entire search space. Genetic algorithm (GA) [11], a common algorithm in the EBAs class, is inspired by generational reproduction and employs imitation crossover, mutation, and elitism to produce new generations to find the global optimum. Differential evolution (DE) [12] is another algorithm inspired by natural evolution that differs from GA in generating next-generation selection operations. The FACDE algorithm integrates Fuzzy C-means clustering, adaptive crossover and cluster-specific mutation strategies to enhance the performance of the DE algorithm [13] and solve the complex water distribution network (WDN) problems [14]. Popular EBAs include the cooperative co-evolutionary algorithm (CCA) [15], the tabu search (TS) [16], and the black widow optimizer algorithm (BWO) [17], among others.

In addition, the SBAs allow social organisms such as insects, animals, and birds to exchange information between several search agents by constructing a multi-agent system in order to discover the global optimal solution. Particle swarm optimization (PSO) [18], the most well-known algorithm in the SBAs category, is modeled after the social behavior and movement of birds in nature. It appears to employ mutual communication and learning among particles to discover the optimal solution within the search space. Phasor particle swarm optimization (PPSO), inspired by phasor theory, is based on modeling the particle control parameters with a phase angle to improve the performance of the PSO algorithm [19]. Ant colony optimization (ACO) [20] primarily simulates ant foraging behavior by having one ant use the soil to demonstrate how to produce pheromones, which other ants then copy in order to find the optimal solution. In recent years, a variety of this type of algorithms have been widely proposed, such as the grey wolf optimizer (GWO) [21], whale optimization algorithm (WOA) [22], salp swarm algorithm (SSA) [23], harris hawks optimization (HHO) [24], moth-flame optimization (MFO) [25], golden eagle optimizer (GEO) [26], slime mould algorithm (SMA) [27], seagull optimization algorithm (SOA) [28], sooty tern optimization algorithm (STOA) [29], sand cat swarm optimization (SCSO) [30], rat swarm optimizer (RSO) [31], and dung beetle optimizer (DBO) [32].

The PBAs usually simulate the physical laws underlying a wide range of natural phenomena, including electromagnetic force, inertial force, light diffraction, reflection, and so on. The few of the famed PBAs are gravitational search algorithm (GSA) [33], specular reflection optimization algorithm (SRA) [34], chaotic multi-specular reflection optimization algorithm considering shared nodes (CMSRAS) [35], equilibrium optimizer (EO) [36], Young's double-slit experiment optimizer (YSDE) [37], Kepler optimization algorithm (KOA) [38], light spectrum optimizer (LSO) [39], Fick's Law Algorithm (FLA) [40], and multi-verse optimizer (MVO) [41], and Turbulent Flow of Water-based Optimization (TFWO) [42].

Humans are widely regarded as the most intelligent creatures, employing a wide range of behaviors and activities, such as teaching, learning, and cooking, to provide optimal solutions to various problems. Many researchers have proposed HBAs inspired by various human behaviors, such as teaching-learning-based optimization (TLBO) [43], socio evolution learning optimization (SELO) [44], poor and rich optimization (PRO) [45], chef-based optimization algorithm (CBOA) [46], teamwork optimization algorithm (TOA) [47], city councils evolution (CCE) [48], and sewing training-based optimization (STBO) [49].

MBAs, which are inspired by mathematical concepts and principles, and GBAs, which are inspired by game rules, player, coach, and referee behavior, are the latter two types of algorithms. Table 1 lists a variety of these MBAs and GBAs.

Exploration and exploitation are the two core elements of metaheuristic algorithms [62], [63]. The exploration is known as global optimization or diversification. Meanwhile, the exploitation is called local optimization or intensification. Metaheuristic algorithms can use the exploration to identify new search space regions and avoid being trapped in local solutions. The exploitation enables metaheuristic algorithms to focus on a specific area in order to discover the optimal solution.

Every metaheuristic algorithm should discover the optimal balance between diversification and intensification; otherwise, the quality of the identified solutions drops. Too many exploration implementations might waste a lot of time and energy. The algorithm simply jumps from one location to another without focusing on searching for higher-quality solutions. The algorithm may be trapped in local optimums and converge prematurely by using excessive exploitation implementations. The sensitivity to fine-tuning of governing parameters is the primary shortcoming of metaheuristic

TABLE 1. Summarization of various MBAs and GBAs.

Category	Name, abbreviation, and references	Year	Inspiration
MBAs	Sine cosine algorithm (SCA) [50]	2016	The sine and cosine functions
	Gradient-based optimizer (GBO) [51]	2021	The gradient-based Newton's method
	Arithmetic optimization algorithm (AOA) [52]	2021	The distribution behavior of the main arithmetic operators
	RUNge Kutta optimizer (RUN) [53]	2021	Runge Kutta method
	weIghted meaN oF vectOrs (INFO) [54]	2022	Weighted mean of vectors
	Exponential distribution optimizer (EDO) [55]	2023	The exponential probability distribution model
	Hybrid average subtraction and standard deviation based optimizer (HASSO) [56]	2023	The amount of information gathered from averages, subtraction, standard deviation
GBAs	Orientation search algorithm (OSA) [57]	2019	The orientation game
	Darts game optimizer (DGO) [58]	2020	The darts game
	Football game based optimization (FGBO) [59]	2020	The behavior of the players during playing the football game
	Ring toss game-based optimization (RTGBO) [60]	2021	The behavior of players and rules of the ring toss game
	Running city game optimizer (RCGO) [61]	2023	The participants' behavior during playing the running city game

algorithms. Additionally, convergence to the global optimum isn't always a given.

As a popular meta-heuristic algorithm, the SCSO algorithm is widely used to solve various optimization problems, such as the feedback controller design [64], the multi-label classification [65], and the process parameter optimization design of the melt-blown [66]. However, The SCSO algorithm has several drawbacks: (i) During the search process, the quality and diversity of sand cat individuals are low, and there is a lack of mutual communication among sand cat individuals. And the local stagnation appears in the late search period to reduce the global search performance of the SCSO algorithm. (ii) The random angle is used to realize the conversion between exploration and exploitation, but the exploitation may begin late, resulting in low convergence efficiency, especially for large-scale high-dimensional multi-mode problems. (iii) The global optimization accuracy and efficiency is weak due to the weakness balance between exploration and exploitation of the SCSO algorithm.

For these existing issues, scholars have really been engaged in enhancing the search ability of the SCSO algorithm. To improve global optimization capability and search efficiency, the modified sand cat swarm optimization (MSCSO) approach, incorporated in the wandering strategy and lens opposition-based learning strategy [67], was presented. The nonlinear periodic adjustment, the pseudo-opposition and pseudo-reflection learning, and elite collaborative mechanisms were introduced into the SCSO algorithm to improve the global convergence ability [68]. The discretized and modified SCSO algorithm based on the mutation concept of the GA was proposed to improve the quality of clustering for the software module system [69]. The PSCSO method, simulated by the political system, was presented to solve complex issues [70]. Combined with the exploration strategy, the hybrid SCSO algorithm was used to solve feature selection [71]. The SCSO algorithm's search capability was improved using the reinforcement learning mechanism [72]. Obviously, only one of these shortcomings is addressed by the techniques presently employed in the literature.

Despite the effectiveness of traditional and newer variants of the SCSO algorithm, none of them can ensure that the global optimum will be found for all optimization situations. The No Free Lunch (NFL) theory [9] has rationally demonstrated this. Numerous scholars were inspired by this theorem to create a new algorithm and develop more effective solutions for new types of problems. Motivated by the above discussions, this paper proposes an intensified SCSO with multiple strategies (IMSCSO) to address some complex optimization problems. In the IMSCSO algorithm, the dynamic random search strategy is initially implemented to increase the algorithm's convergence efficiency. Additionally, a hybrid opposition-based learning strategy is developed to boost population diversity and avoid premature convergence of the algorithm. Lastly, the joint opposite selection strategy is produced to strike a balance between exploration and exploitation of the algorithm. Correspondingly, the main contributions of this paper are outlined below:

(1) A new hybrid opposition-based learning technique is created based on the lens opposition-based learning and quasi-oppositional learning to improve the exploitation ability of the IMSCSO algorithm.

(2) An intensified SCSO with multi-strategies (IMSCSO) is proposed that contains the dynamic random search mechanism, hybrid opposition-based learning mechanism, and joint opposite selection mechanism.

(3) The IMSCSO algorithm is comprehensively evaluated through a series of mathematical test functions (including both 23 benchmark functions, 29 CEC 2017 benchmark

functions, 10 CEC 2019 benchmark functions, and 10 CEC 2020 benchmark functions), and the experimental results indicate that the IMSCSO algorithm has a more competitive performance compared with other state-of-the-art optimizers.

(4) The MSCSO algorithm has been used effectively in various practical engineering design issues, demonstrating that it is promising for solving real-world application difficulties.

The rest of this paper is arranged as follows. Section II briefly presents the SCSO algorithm. Section III introduces the developed IMSCSO in detail. Section IV displays the experiments and discussion. Section V provides the results on seven engineering optimization scenarios. Conclusions and future works are listed in the last section.

II. SCSO ALGORITHM

A. INITIALIZATION PHASE

In the initial phase of the SCSO algorithm, the population size of sand cats and the dimension size of the optimization problem are defined as N and D, respectively. As seen in Figure 1, a sand cat is a 1×D array representing the fitness of the optimization problem. The sand cat population is described by Eq. (1), and the position of each sand cat in all dimensions can be represented by Eq. (2). In addition, the fitness function of the problem is evaluated on Eq. (3). In each iteration, this fitness function evaluation is carried out for each cat and continually updates to obtain the best fitness value in the following iteration.

$$X^{i}_{Sand\ cats} = \{SC_1, SC_2, \cdots, SC_N\}, 1 \le i \le N \quad (1)$$

$$X^{i}_{j\,Sand\ cats} = \{SC_1, SC_2, \cdots, SC_N\}, \quad 1 \le i \le N, 1 \le j \le D \quad (2)$$

$$Fitness(X^{i}_{Sand\ cats}) = f(SC_1, SC_2, \cdots, SC_N), \quad \forall Xi(\text{is evaluatedfor } m\text{time}) \quad (3)$$

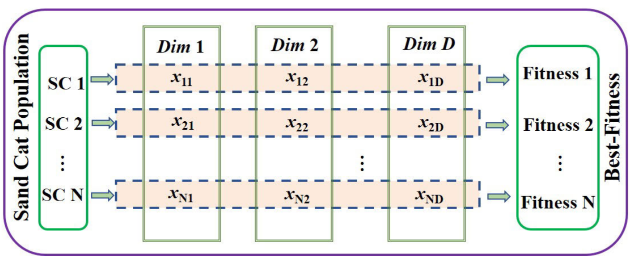

FIGURE 1. Initialization phase of the SCSO algorithm.

B. PREY-SEEKING STRATEGY (EXPLORATION)

The prey-seeking strategy of sand cats uses low-frequency noise emission to continuously update the location of sand cats to explore the prey in the search space, which improves the exploration ability of the SCSO algorithm. It is proven mathematically as indicated in Eqs. (4)-(7).

$$\vec{r_c} = S - \left(\frac{S \times m}{M_{\max}}\right) \quad (4)$$

$$\vec{R} = 2 \times \vec{r_c} \times \text{rand}(0, 1) - \vec{r_c} \quad (5)$$

$$\vec{r_b} = \vec{r_c} \times \text{rand}(0, 1) \quad (6)$$

where, $\vec{r_c}$ is a linearly decreasing sensitivity factor within [2,0]; S is set to 2; m and M_{max} are the current iterations and the maximum iterations, respectively. $\vec{r_c}$ is a control factor; $\vec{r_b}$ is sensitivity range; $\vec{R}$ is balanced factor between exploration and exploitation.

$$\vec{X}(m+1) = \vec{r_b} \cdot (\vec{X_b}(m) - rand(0, 1) \cdot \vec{X_s}(m)) \quad (7)$$

where, $\vec{X_b}$ and $\vec{X_s}$ are the best position of sand cats and the current position of sand cats, respectively.

C. PREY-ATTACKING STRATEGY (EXPLOITATION)

In the SCSO algorithm, the sand cat attacks prey by moving randomly according to its own sensitivity range. Their position is continually updated by utilizing the adaptive attack angle α to approach the position of the prey. Conveniently, the prey-attacking strategy is demonstrated mathematically as stated in Eqs. (8) and (9). The SCSO algorithm exhibits a noteworthy balanced search behavior by utilizing the adaptive parameter R to effectively coordinate both exploration and exploitation, as demonstrated in Equation (10).

$$\overrightarrow{X_{rand}} = \left| rand(0, 1) \cdot \vec{X_b}(m) - \vec{X_s}(m) \right| \quad (8)$$

$$\vec{X}(m+1) = \vec{X_b}(m) - \vec{r_b} \cdot \overrightarrow{X_{rand}} \cdot \cos(\alpha) \quad (9)$$

$$\vec{X}(m+1) = \begin{cases} \vec{X_b}(m) - \vec{r_b} \cdot \overrightarrow{X_{rand}} \cdot \cos(\alpha) & |R| \le 1(a) \\ \vec{r_b} \cdot (\vec{X_b}(m) - rand(0, 1) \cdot \vec{X_s}(m)) & |R| > 1(b) \end{cases} \quad (10)$$

where, $\overrightarrow{X_{rand}}$ is the random position of sand cats; α is a random angle of the movement direction within $[0,2\pi]$. The pseudocode of SCSO is summarized in **Algorithm 1.**

Algorithm 1 SCSO Pseudocode

1. Initialize the population (sand cats).
2. Calculate the fitness function according to the objective function.
3. **While** ($m <= M_{max}$)
4. Calculate the $\vec{r_c}$, $\vec{R}$, $\vec{r_b}$ by Eqs. (4), (5), and (6), respectively.
5. Determine the random angle α by the Roulette Wheel Selection method.
6. **For** each sand cat
7. **If** (abs ($\vec{R}$) <=1)
8. Update the position of sand cats by Eq. 10a.
9. **Else**
10. Update the position of sand cats by Eq. 10b.
11. **End if**
12. **End for**
13. $m = m{+}{+}$
14. **End while**
15. **Return**the best fitness.

Algorithm 2 Dynamic Random Search Strategy

1. Initialization parameters: the termination criteria E, maximum number of iterations T, the step factor α_0, *epoch*=0, $j = 0$, $X_{current} = X_b$, the fitness function $F(X)$.
2. Set $t = 0$.
3. Generate a random vector dx within [-α_j, α_j].
4. Let *epoch*= *epoch*+1.
5. $F_{New} = F\ (X_{current} + dx)$.
6. **If** $F_{New} < F_{Best}$
7. $X_b = X_{current} + dx$, $F_{Best} = F_{New}$, $t = t + 1$. Return to step 19.
8. **End if**
9. **If** $F_{New} < F_{Current}$
10. $X_{current} = X_{current} + dx$, $F_{Current} = F_{New}$, $t = t + 1$. Return to step 19.
11. **End if**
12. $F_{New} = F\ (X_{current} - dx)$.
13. **If** $F_{New} < F_{Best}$
14. $X_b = X_{current} - dx$, $F_{Best} = F_{New}$, $t = t + 1$. Return to step 19.
15. **End if**
16. **If** $F_{New} < F_{Current}$
17. $X_{current} = X_{current} - dx$, $F_{Current} = F_{New}$, $t = t + 1$. Return to step 19.
18. **End if**
19. **If**$t < T$ and then return to step 3.
20. $j = j + 1$, $\alpha_j = \alpha_{j-1} * 0.5$.
21. If $epoch = E$, and then stop the iteration. Otherwise, return to step 2.

III. THE PROPOSED IMSCSO ALGORITHM

A. DYNAMIC RANDOM SEARCH STRATEGY

Based on the general search stage and the local search stage, the dynamic random search technique (DRST) can find the global optima only when given enough time [73]. DRST obtains the current optimal solution by using the general search (GS) strategy, followed by the local search (LS) strategy to conduct a fine search around the current optimal solution to obtain a better solution, effectively avoiding the adaptive random search technique [74] from falling into a local optimum. As a result, the LS technology in the DRST is used to perform perturbation and mutation of the present global optimal solution (X_b), as well as to improve the exploitation ability, search accuracy, and search efficiency of the SCSO algorithm at the late iteration. The pseudocode of the LS technology in the DRST is listed in **Algorithm 2**.

B. HYBRID OPPOSITION-BASED LEARING STRATEGY

By allowing search agents to search in the opposite way, opposition-based learning (OBL) [75] increases the number of candidate solutions, boosts the probability that the algorithm will discover the global best solution, and essentially strengthens the algorithm's search capabilities. However, OBL can only produce the opposite solutions at specific locations, making it difficult to adequately boost the algorithm's search efficiency for complex optimization problems. Abundant OBL variants have been suggested in recent years, including lens opposition-based learning (LOBL) [76] and quasi-oppositional learning (QOL) [77]. LOBL, inspired by the lens imaging principle, is proposed to enrich the population diversity and search efficiency of the algorithm, as shown in Figure 2a. Based on the OBL, the QOL is proposed to broaden the search range of the algorithm to obtain the best solution among candidate solutions and quasi-opposite solutions [78], [79], as illustrated in Figure 2b. Combining the advantages of LOBL and QOL, a hybrid opposition-based learning (HOBL) strategy is proposed to improve the search performance of the SCSO algorithm, as shown in Figure 2c. As displayed in Figure 2, the LOBL, QOL, and HOBL are described first below in detail.

1) LOBL MECHANISM

In Figure 2a, the cardinal point O is the midpoint of the search range [LB, UB]. The object P_2 is generated by the object P_1 through a convex lens. H_1 and H_2 are the heights of the objects P_1 and P_2, respectively. Meanwhile, the x_i ($i = 1,2,\ldots, N$) (i.e., the *i-th* solution of the search agents) and x_{LOBLi} (the *i-th* opposite solution of the search agents) are the projections of the objects P_1 and P_2 at the axis, respectively. Consequently, Eq. (11) is calculated as follows.

$$\frac{(LB + UB)/2 - x_i}{x_{LOBLi} - (LB + UB)/2} = \frac{H_1}{H_2} \tag{11}$$

Let, $l = H_1/H_2$, obviously, the opposite solution x_{LOBL} is obtained by Eq. (12).

$$x_{LOBLi} = \frac{(LB + UB)}{2} + \frac{(LB + UB)}{2l} - \frac{x_i}{l} \tag{12}$$

2) QOL MECHANISM

In Figure 2b, the point S_1 (x_i) is located between [LB, UB], and the points S_2 (x_o) and S_3 (x_{qo}) are the opposite point and

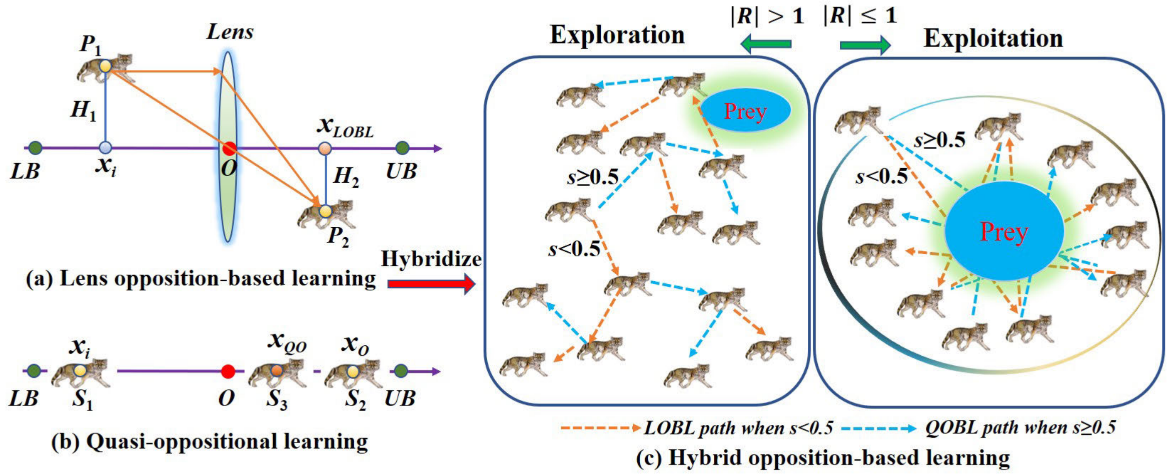

FIGURE 2. Hybrid opposition-based learning.

quasi-opposite point of the point S_1 based on the midpoint O between $[LB, UB]$, respectively. Therefore, the opposite point x_o and the quasi-opposite point x_{qo} are completely obtained by Eqs. (13) and (14), respectively.

$$x_{Oi} = (LB + UB) - x_i \tag{13}$$

$$x_{qoi} = rand\left[\left(\frac{LB+UB}{2}\right), (LB+UB-x_i)\right] \tag{14}$$

3) HOBL MECHANISM

Based on the LOBL and QOL mechanism, the HOBL mechanism is proposed by a random switching probability (s). The HOBL is given as follows.

$$x_{\text{HOBL } i} = \begin{cases} \frac{(LB+UB)}{2} + \frac{(LB+UB)}{2l} - \frac{x_i}{l} & \text{if } s < 0.5 \\ \text{rand}\left[\left(\frac{LB+UB}{2}\right), (LB+UB-x_i)\right] & \text{else} \end{cases} \tag{15}$$

where, x_i represents the *i-th* solution of the search agents, x_{HOBL} represents the *i-th* opposite solution of x_i obtained by HOBL, l represents the distance factor, s represents the random switching probability within [0, 1], LB and UB are the lower and upper boundary of the search agents.

Generally, the HOBL can also be generalized to the D-dimensional space as follows.

$$x_{HOBLi,j} = \begin{cases} \frac{(LB_j+UB_j)}{2} + \frac{(LB_j+UB_j)}{2l} - \frac{x_{i,j}}{l} & if\ s < 0.5 \\ rand\left[\left(\frac{LB_j+UB_j}{2}\right), (LB_j+UB_j-x_{i,j})\right] & else, \\ j = 1, 2, \cdots, D \end{cases} \tag{16}$$

where, $x_{i,j}$ and $x_{HOBLi,j}$ represents the j-dimensional components of x_i and x_{HOBLi}, respectively. LB_j and UB_j are the lower and upper boundary of the search agents in j-dimensional space.

In the SCSO algorithm, because the search information of sand cats is not effectively shared with other sand cats when near the optimal solution, it can lead to a decrease in population diversity and be prone to falling into a local optimum. The HOBL mechanism is introduced into the SCSO algorithm to expand the population's diversity in the exploration and exploitation phases (as seen in Figure 2c) and strengthen the search ability of the algorithm. In Figure 2c, based on the HOBL mechanism, sand cats can expand the search area and enrich the number of search agents to discover prey with faster search efficiency, so as to enhance the exploration ability of the algorithm. When the sand cats discover the prey, they frequently change their attack position using the HOBL mechanism to capture the prey and boost the exploitation ability of the algorithm.

C. JOINT OPPOSITE SELETION

Joint Opposite Selection (JOS) is a learning technology based on joint opposition, which combines the advantages of dynamic opposition (DO) and selective leading opposition (SLO), effectively balancing exploration and exploration, and improving the performance and efficiency of algorithms in the search space [80], [81].

1) DO MECHANISM

DO mechanism [82] is proposed by combing the QOL with quasi-reflection-based learning (QRBL) [83] to dynamically expand search space by using the opposite strategy, which prevents the algorithm from falling into a local optimum. The specific calculation is as follows.

$$x_{i,j}^{o} = \text{rand} \times \left(LB + UB - x_{i,j}\right),\ i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, D \tag{17}$$

$$x_{i,j}^{DO} = x_{i,j} + rand \times \left(x_{i,j}^{O} - x_{i,j}\right),\ \text{if rand } < jr\quad i = 1, 2, \ldots, N, j = 1, 2, \ldots, D \tag{18}$$

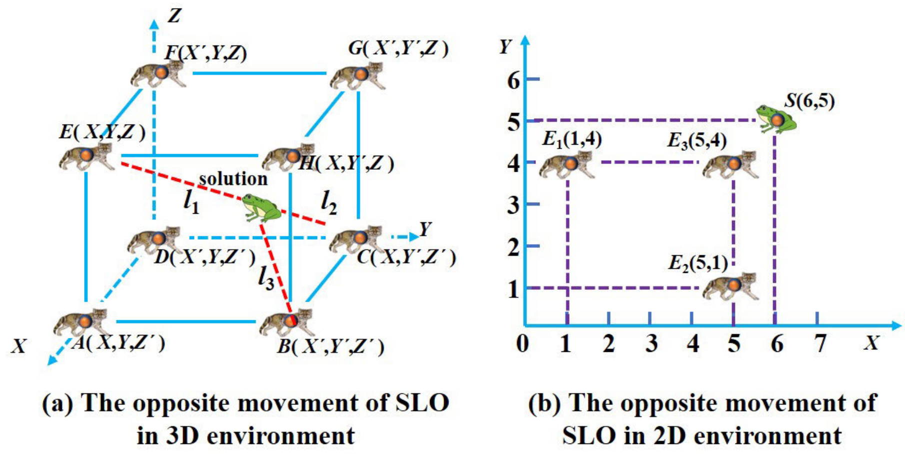

FIGURE 3. The opposite movement of SLO in 2D and 3D environment.

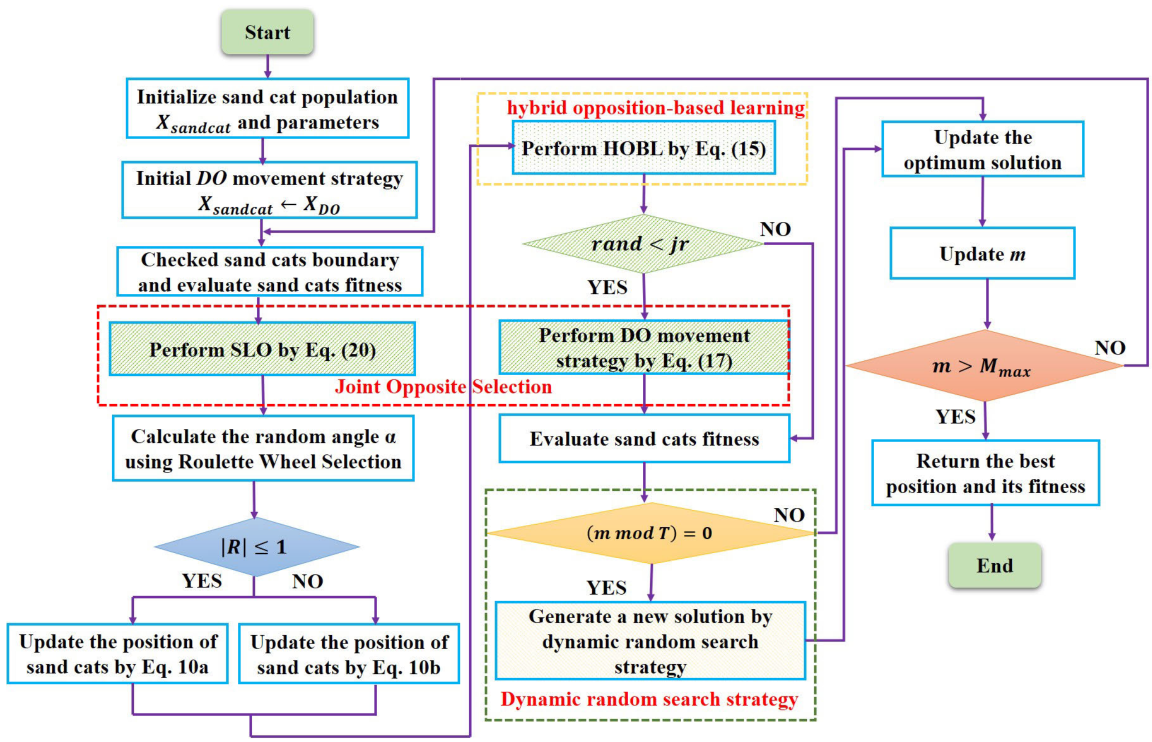

FIGURE 4. The flowchart of the IMSCSO algorithm.

where, $x_{i,j}^{O}$ is the opposite solution and dynamic opposite solution of $x_{i,j}$, respectively. jr is the jump rate.

2) SLO MECHANISM

Inspired by Selective Opposition (SO), the SLO strategy idea [84] is to calculate the difference distance (ds) between the current solution and the optimal solution in each dimension and compare it with the set threshold. When $ds >$ threshold, the dimension is regarded as the faraway distance dimension (df). Conversely, the dimension is regarded as the close distance dimension (dc) Further, the Spearman's Rank Correlation Coefficient (src) between the current solution and the optimal solution can be calculated. If $dc > df$, $src \le 0$, the SLO strategy is executed.

The opposite movement of SLO in 2D and 3D environment is illustrated in Figure 3. In Figure 3a, the point $B\,(X', Y', Z')$ is the exact opposite point of the leading point of the search agent $E(X, Y, Z)$, and the distances between the points E and B to the solution are l_1 and l_3 in turn. Additionally, the last two dimensions of $E\,(X, Y, Z)$ are opposed to $C\,(X, Y', Z')$ with the distance l_2. The distance l_2 is close to the solution. In Figure 3b, the coordinate X and Y are the first and second dimension, successively. E_1, E_2 and E_3 represent the coordinate position of sand cats, and S represent the coordinate position of a frog. The difference distances between E_2 and S are 1 and 4 at the first and second dimension respectively. Let the threshold is equal to 3, this difference distance is less than 3. That means the coordinate X of this sand cat is closer to the frog than other sand cats. In addition, this closer sand cat is selected as the leader for preying. Lastly, the SLO strategy will implement at the first dimension. However, the difference distance of the second dimension is greater than 3; it means the SLO strategy will not apply. Similarly, for the coordinate position E_1, SLO strategy should be applied to the second dimension. For the coordinate position E_3, SLO strategy should be applied to the first and second dimensions. The SLO strategy is calculated as follows.

$$dd_{i,j}(m) = \left|x_{b,j}(m) - x_{i,j}(m)\right| \tag{19}$$

$$\text{src} = \frac{6 \times \sum_{j=1} \left(dd_{i,j}\right)^2}{dd_{i,j} \times \left(dd_{i,j}^2 - 1\right)} \tag{20}$$

$$x'_{i,d\ c} = LB + UB - x_{i,d\ c}, \text{ if } src \leq 0 \text{ and } dc > df \tag{21}$$

where, $x_{i,dc}$ is the closer dimension of the i-th solution. $x'_{i,dc}$ is the opposite solution of $x_{i,dc}$. src represents the Spearman's Rank Correlation Coefficient. $dd_{i,j}$ is the difference distance.

In the SCSO algorithm, when the sand cats adopt the JOS strategy, the SLO strategy assists the sand cats to succeed in exploitation phase by changing their close distance dimension, and the DO strategy tries to diverse the search space range of the sand cats in the exploration phase. The SCSO with JOS strategy can boost efficiency and balance between exploration and exploitation.

D. THE PSEUDOCODE AND FLOWCHART OF IMSCSO

Pseudocode and flowchart of IMSCSO are shown in **Algorithm 3** and Figure 4, respectively.

IV. COMPUTATIONAL COMPLEXITY OF IMSCSO

The main three components of SCSO are initialization, fitness assessment, and updating of sand cats. The computational complexity of SCSO is $O(N \times (M_{max} + M_{max} \times D+1))$, that is, $O(N)$. Where N is the number of sand cats, M_{max} is the maximum number of iterations, and D is the dimension of specific problems. The computational complexity of JOS is $O(SLO)+O(DO)$, which is $O(N \times M_{max} \times D_C) + O(N \times M_{max} \times D \times jr)$. The computational complexity of HOBL is $O(2N \times M_{max} \times D)$. The computational complexity of DRST is $O(E \times M_{max} \times D)$. Where D_C is the close distance dimension, jr is jumping rate, E is the maximum number of iterations by LS. Therefore, the computational time of IMSCSO is $O(N \times M_{max}(2+D_C + D(M_{max}+jr))+E \times M_{max} \times D)$, that is, $O(N)$. Hence, the computational complexity of IMSCSO has the same order-time complexity as that of SCSO.

Algorithm 3 IMSCSO Pseudocode

1. Initialize the population size N, the maximum iterations M_{max}.
2. Initialize the random population $X_{sandcat}$.
3. Generate initial random population of X_{DO} based on $X_{sandcat}$.
4. $X_{sandcat} \leftarrow X_{DO}$.
5. **While** $(m <= M_{max})$
6. Checked boundary $X_{sandcat}$, calculate the fitness value of each sand cat, and obtain the best solution.
7. Update Position of $X_{sandcat}$.
8. Set selective boundary and $\vec{r}_c$ as the threshold for SLO.
9. Perform SLO for each sand cat by Eq. (21).
10. Update $\vec{r}_c$, $\vec{R}$, $\vec{r}_b$ by Eqs. (4), (5), and (6), respectively.
11. Determine the random angle α by the Roulette Wheel Selection method.
12. **For** each sand cat
13. **If** (abs $(\vec{R})$ <=1)
14. Update the position of sand cats by Eq. 10a.
15. Update the position of sand cats by Eq. (16).
16. **Else**
17. Update the position of sand cats by Eq. 10b.
18. Update the position of sand cats by Eq. (16).
19. **End if**
20. **End for**
21. **If**$rand<jr$
22. Perform DO position X_{DO} by Eq. (18), $X_{sandcat} \leftarrow X_{DO}$.
23. **End if**
24. Evaluate the fitness values of each sand cat.
25. **If**$(mmodT) = 0$
26. Generate a new solution by dynamic random search strategy.
27. **Else**
28. Update the best fitness.
29. **End if**
30. $m = m+1$.
31. **End while**
32. **Return**the best fitness.

V. EXPERIMENTS AND DISCUSSIONS

This section evaluates the performance of the IMSCSO algorithm by employing five recent challengeable mathematical test suites: 23 classical test functions (as seen in Table 16), CEC 2017 (as seen in Table 17) [85], CEC 2019 (as seen in Table 18) [86], and CEC 2020 (as seen in Table 19) [87]. We chose eleven of the most widely used metaheuristic algorithms currently available as comparison objects: GWO [21], WOA [22], HHO [24], SCSO [30], KOA [38], SCA [50], AOA [52], PSO [18], DBO [32], SOA [28], MVO [41], DO [99], and SWO [102].

Table 2 displays the parameter settings for the aforementioned algorithms. All experiments are run with MATLAB 2020a, Intel Xeon Silver 4110 2.1 GHz, RAM 32GB, and

64-bit Windows 10 professor. Since the algorithms under investigation are all stochastic, in this paper, the maximum number of iterations is set to be 1000, each function is repeated 30 times, and the population size of all algorithms is set to be 50. The mean value and standard deviation (SD) are selected as an evaluation metric. Additionally, the Friedman test [100] and Wilcoxon rank-sum test [101] are utilized to determine whether there is a difference and superiority between the IMSCSO's outcomes and those of the competing optimizers.

TABLE 2. Parameter settings of the compared optimizers and proposed IMSCSO.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
GWO	a	[2,0]	HHO	E_0 J	[-1,1] [0,2]
WOA	a Spiral factor b	[2,0] 1	SCA	A	2
SCSO	r_G R	[2,0] [$-2r_G$, $2r_G$]	KOA	T u_0 γ	3 0.1 15
AOA	α u	5 0.05	DO	α k	[0,1] [0,1]
IMSCSO	E T Jr	100 3*Dim 0.2	PSO	c_1 c_2 V_{max}	2 2 6
DBO	k λ b S	0.1 0.1 0.3 0.5	SOA	A f_c	[2,0] 2
SWO	TR CR	0.3 0.2	MVO	WER TDR	[0.2,1] [0.6,0]

A. PARAMETER SENSITIVITY ANALYSIS

The proposed IMSCSO algorithm employs four parameters, i.e., number of sand cats (N), maximum number of iterations (M_{max}), jumping rate (Jr), and parameter T. By changing these parameters' values while holding the other parameters constant, the sensitivity analysis of the proposed IMSCSO algorithm has been examined.

(1) Number of sand cats (N): IMSCSO algorithm was simulated for different values N, The values of N used in experimentation are 50, 100, 150, and 200. Figure 5a shows the variations of different number of sand cats on benchmark test functions. As the number of sand cats (N) increases, it can be seen from Figure 5a that the fitness function's value drops.

(2) Maximum number of iterations (M_{max}): IMSCSO algorithm was run for by different values of M_{max}. The values of M_{max} are set to 200, 400, 800, and 1200. The impact of M_{max} on benchmark test functions is depicted in Figure 5b. The findings demonstrate that as the number of iterations is increased, IMSCSO converges towards the best solution.

(3) Variation in parameter Jr: The IMSCSO algorithm was performed for various values of Jr while keeping the other parameters constant in order to explore the effect of the parameter Jr. The parameter Jr is run from 0.05 to 0.95 with a 0.05 increment. Figure 6a show the variation of Jr on UF7, MF12, and MF13 functions, respectively. The results indicate that IMSCSO produces better optimal outcomes when the value of Jr is set to 0.2.

(4) Variation in parameter T: To investigate the impact of the parameter T, the IMSCSO algorithm was run for a range of T values while holding the other parameters constant. Here, the Dim is the dimension of benchmark test functions. The parameter $T$is varied from $1{*}Dim$ to $*Dim$ with a $1{*}Dim$ increment. Figure 6b show the variation of T on UF7, MF12, and MF13 functions. Obviously, for the purpose of simplicity, all optimization problems in our studies use $3{*}Dim$ as the size of T.

B. EXPLOITATION ANALYSIS OF IMSCSO

In this section, we evaluate the exploitation ability of the IMSCSO algorithm on unimodal test functions (UF1-UF7). Table 3 provides IMSCSO and other algorithms' results on unimodal functions. The obtained values demonstrate the IMSCSO's superiority, as it outperforms its competitors in reaching the lowest values of the mean values with excellent reliability and consistency (minimum SD) for six functions (i.e., UF1, UF2, UF3, and UF4) in this collection. For function UF5, the mean and SD values of the IMSCSO are superior to GWO, WOA, SCSO, AOA, SCA, KOA, DO, and SOA. For function UF7, IMSCSO can provide the second-best results compared to other algorithms. For function UF6, the mean value of the IMSCSO is less than WOA, HHO, DO, PSO, DBO, and MVO. The SD value of IMSCSO is superior to KOA, SWO, SOA, and SCSO. This observation demonstrates the substantial exploitation capability of the IMSCSO in the intensification. This is due to the fact that IMSCSO adopts the DRST and HOBL strategies to be primarily focused on increasing the diversity of the sand cats and widening the local search area. This promotes the search engine's optimization process to narrowly and broadly target the local area.

C. EXPLORATION ANALYSIS OF IMSCSO

In this section, the exploration capability of the IMSCSO is examined using sixteen multimodal functions (MF8-MF23). The results obtained for the multimodal function between IMSCSO and the other methods are illustrated in Table 4. IMSCSO obtains minimum values of mean and SD for eight functions, as demonstrated in the functions MF9, MF10, MF11, MF12, MF15, MF18, MF20, and MF21. In addition, for functions MF14, MF16, MF17, and MF19, IMSCSO achieves the minimum or same mean value compared to the other methods. It can be seen that IMSCSO algorithm has a better exploration capability and competitive on most multimodal functions. The reason is that IMSCSO has the prey-seeking strategy of sand cats with the HOBL mechanism and JOS strategy, which are helpful in expanding the exploration search area and narrowing the exploitation search area.

D. PERFORMANCE EVALUATION ON CEC-2017 TEST FUNCTIONS

In Appendix A (Table 17), popular and challenging CEC-2017 benchmark issues were employed to further analyze the performance of the IMSCSO algorithm. These challenges

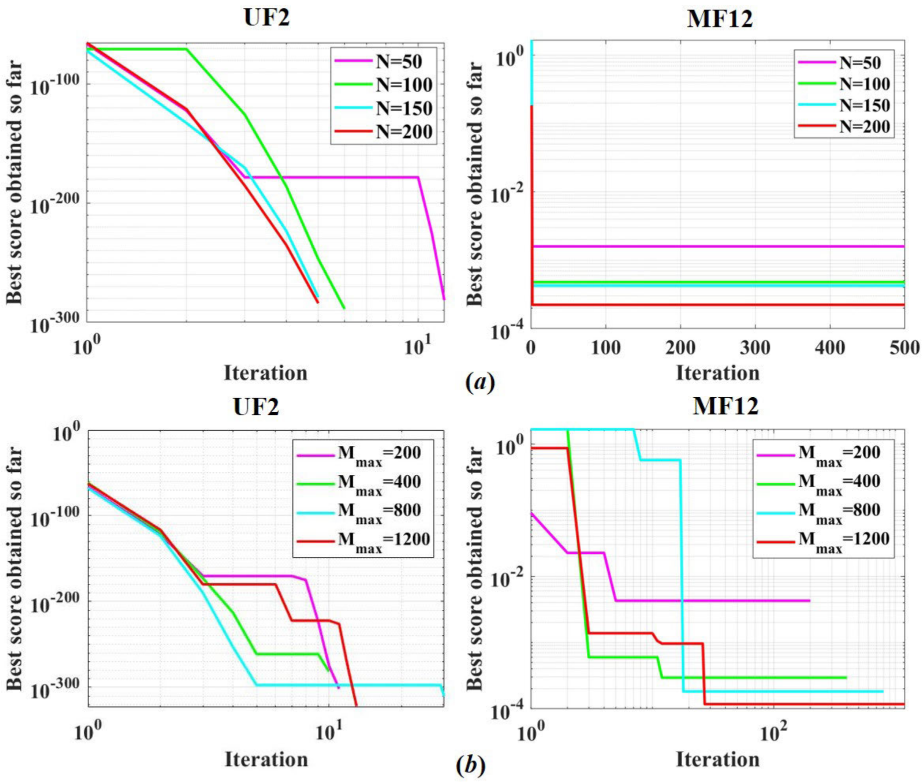

FIGURE 5. Sensitivity analysis of IMSCSO algorithm: (a) *N*, and (b) *Mmax*.

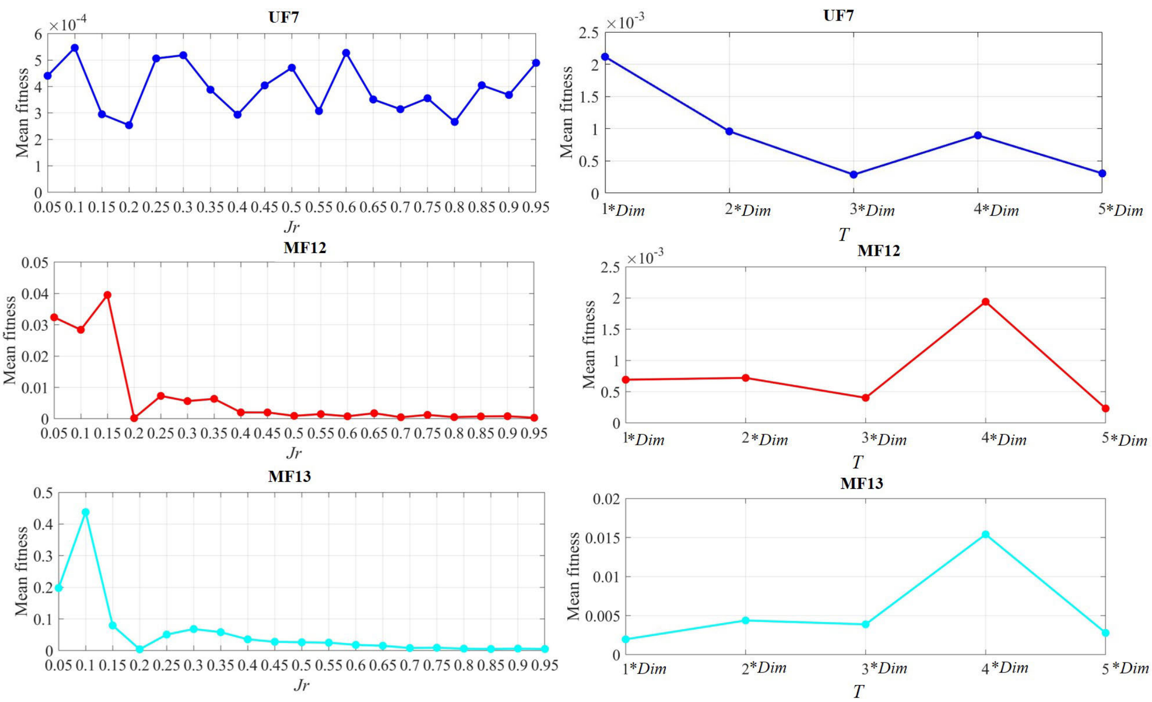

FIGURE 6. Sensitivity analysis of IMSCSO algorithm: (a) Jr, and (b) T.

included rotated and shifted unimodal, multimodal, hybrid, and composite test functions [85]. The results obtained by INFO and the other well-known methods are reported in Table 5. In Table 5, compared to other algorithms, it is showed

TABLE 3. Results of unimodal functions.

Algorithms	UF1		UF2		UF3		UF4	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	3.99E-70	1.45E-69	6.04E-41	6.88E-41	1.03E-19	2.78E-19	1.70E-17	2.17E-17
WOA	2.51E-170	**0.00E+00**	1.93E-109	5.27E-109	1.01E+04	7.52E+03	2.91E+01	3.02E+01
SCSO	6.17E-238	**0.00E+00**	1.47E-126	4.60E-126	6.96E-206	0.00E+00	3.62E-103	1.84E-102
AOA	3.51E-45	1.92E-44	**0.00E+00**	**0.00E+00**	2.68E-04	1.02E-03	1.33E-02	1.92E-02
HHO	2.59E-191	**0.00E+00**	1.66E-100	8.24E-100	1.34E-153	7.36E-153	1.68E-96	9.10E-96
SCA	8.54E-04	2.02E-03	4.81E-06	1.13E-05	2.98E+03	2.66E+03	1.40E+01	8.82E+00
KOA	1.27E+04	2.63E+03	6.11E+01	1.82E+01	3.63E+04	1.16E+04	4.96E+01	5.15E+00
DO	7.74E-10	4.98E-10	2.16E-05	1.01E-05	8.91E-02	5.37E-02	2.77E-02	2.79E-02
PSO	1.48E-01	7.14E-02	7.34E-01	3.08E-01	5.19E+01	1.56E+01	1.43E+00	2.25E-01
DBO	3.39E-224	**0.00E+00**	1.98E-128	1.04E-127	3.70E-155	2.02E-154	4.96E-116	1.81E-115
SWO	2.59E-02	8.88E-02	1.53E-02	3.60E-02	2.78E+00	5.72E+00	6.52E-02	1.22E-01
SOA	1.83E-15	7.72E-15	2.61E-14	6.11E-14	3.87E+04	1.87E+04	4.92E-37	2.66E-36
MVO	3.02E-01	1.00E-01	1.08E+01	3.35E+01	4.95E+01	1.91E+01	6.60E-01	2.93E-01
IMSCSO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Algorithms	UF5		UF6		UF7	
	Mean	SD	Mean	SD	Mean	SD
GWO	2.64E+01	6.89E-01	3.75E-01	2.52E-01	5.03E-04	3.23E-04
WOA	2.66E+01	4.74E-01	4.50E-03	3.60E-03	1.04E-03	1.17E-03
SCSO	2.76E+01	9.53E-01	1.03E+00	4.18E-01	3.47E-04	3.02E-04
AOA	2.78E+01	3.90E-01	2.46E+00	2.46E-01	**1.77E-05**	**1.45E-05**
HHO	1.12E-03	1.57E-03	1.49E-05	2.00E-05	4.05E-05	3.76E-05
SCA	4.52E+01	3.52E+01	4.18E+00	2.74E-01	1.77E-02	9.90E-03
KOA	9.60E+06	4.38E+06	1.17E+04	2.99E+03	5.29E+00	2.26E+00
DO	2.52E+01	2.71E-01	3.18E-07	1.31E-07	5.42E-03	2.66E-03
PSO	1.73E-13	5.12E-13	1.38E-01	5.78E-02	2.11E+00	1.39E+00
DBO	**0.00E+00**	**0.00E+00**	**9.60E-18**	**3.50E-17**	6.83E-04	7.08E-04
SWO	4.61E-05	5.95E-05	3.47E+00	1.02E+00	1.18E-01	1.72E-01
SOA	1.71E-01	2.31E-01	2.14E+00	9.27E-01	8.06E-05	6.95E-05
MVO	1.28E-05	1.62E-05	1.74E-01	4.21E-02	1.75E-02	5.81E-03
IMSCSO	3.41E-02	1.81E-02	1.79E-01	3.53E-01	3.61E-05	3.31E-05

that the IMSCSO algorithm gets the best results (the minimum mean and SD values) on eight functions (i.e., MF17-5, MF17-8, MF17-10, HF17-12, HF17-13, HF17-14, HF17-19, and CF17-22). The IMSCSO algorithm attains the minimum SD value on functions MF17-3 and HF17-11. For functions MF17-7 and HF17-17, the mean value of IMSCSO is better than other algorithms. It has been concluded that the IMSCSO algorithm attains prominent results for the majority of CEC 2017 test functions. IMSCSO has superior performance compared with other well-known optimizers (i.e., GWO, WOA, SCSO, AOA, HHO, SCA, KOA, DO, PSO, DBO, SWO, SOA, and MVO) when dealing with complicated optimization issues.

E. PERFORMANCE EVALUATION ON CEC 2019 TEST FUNCTIONS

In this section, additional tests are performed on the CEC 2019 test suite to verify the performance of the proposed IMSCSO as well as the performance of the competing algorithms. Table 6 presents the mean and SD values attained by analyzing the results of IMSCSO and other methods. As it can be seen from Table 6, for functions C19-1 and C19-8, the mean and SD values of IMSCSO are superior to others methods. The IMSCSO gets the minimum SD value on functions C19-2, C19-4, and C19-6. For other CEC 2019 test functions, the IMSCSO algorithm also can provide some competitive results. These results demonstrate that the IMSCSO algorithm has a reliable and accurate efficiency for solving CEC 2019, compared to the other optimizers.

F. PERFORMANCE EVALUATION ON CEC 2020 TEST FUNCTIONS

In this section, for proper exploration and exploitation, the IMSCSO algorithm is tested on ten CEC 2020 test functions. The obtained outcomes of the CEC 2020 test functions are provided in Table 7. The results in Table 7 show that IMSCSO achieves the global optimum in eight functions (i.e., UF20-1, HF20-4, HF20-5, HF20-6, HF20-7, CF20-8, CF20-9, and

TABLE 4. Results of multimodal functions.

Algorithms	MF8		MF9		MF10		MF11	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	-6.30E+03	6.18E+02	3.79E-15	1.44E-14	1.42E-14	2.07E-15	1.77E-03	4.82E-03
WOA	-1.18E+04	9.28E+02	1.89E-15	1.04E-14	4.20E-15	1.85E-15	7.85E-04	4.30E-03
SCSO	-7.07E+03	7.43E+02	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**
AOA	-6.14E+03	4.64E+02	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	7.32E-02	4.79E-02
HHO	**-1.25E+04**	1.43E-01	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**
SCA	-4.01E+03	2.58E+02	1.49E+01	2.51E+01	1.44E+01	8.61E+01	1.81E-01	2.37E-01
KOA	-4.07E+03	4.80E+02	2.71E+02	2.02E+01	1.85E+01	1.05E+00	1.03E+02	1.96E+01
DO	-9.20E+03	6.17E+02	1.33E+01	1.22E+01	5.65E-06	1.95E-06	1.53E-02	2.06E-02
PSO	-6.40E+03	1.05E+03	1.16E+02	2.69E+01	1.20E+00	6.23E-01	1.84E-02	1.05E-02
DBO	-9.97E+03	2.23E+03	2.05E+00	7.86E+00	1.01E-15	6.49E-16	**0.00E+00**	**0.00E+00**
SWO	-5.36E+03	4.48E+02	1.69E-02	5.44E-02	9.67E-03	1.76E-02	8.78E-02	1.75E-01
SOA	**-1.25E+04**	**2.22E-05**	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**
MVO	-7.71E+03	7.29E+02	1.21E+02	2.93E+01	1.09E+00	7.38E-01	5.75E-01	9.52E-02
IMSCSO	-8.89E+03	1.37E+03	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Algorithms	MF12		MF13		MF14		MF15	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	2.39E-02	1.32E-02	2.54E-01	1.82E-01	3.21E+00	3.89E+00	3.68E-03	7.59E-03
WOA	8.62E-04	1.44E-03	6.33E-02	6.20E-02	1.78E+00	1.89E+00	6.10E-04	2.72E-04
SCSO	4.08E-02	1.98E-02	2.17E+00	3.53E-01	1.79E+00	1.90E+00	3.38E-04	1.67E-04
AOA	3.15E-01	4.65E-02	2.78E+00	1.41E-01	1.00E+01	3.84E+00	8.14E-03	1.26E-02
HHO	5.82E-07	6.40E-07	**7.37E-06**	**1.16E-05**	**9.98E-01**	8.34E-10	3.81E-04	2.36E-04
SCA	9.71E-01	1.31E+00	4.28E+00	3.47E+00	1.33E+00	7.52E-01	8.44E-04	3.75E-04
KOA	5.13E+06	4.67E+06	2.37E+07	1.54E+07	2.93E+00	1.86E+00	5.21E-03	4.86E-03
DO	2.13E-07	9.38E-08	3.66E-04	2.01E-03	**9.98E-01**	**5.30E-16**	4.61E-04	3.93E-04
PSO	6.46E-03	1.98E-02	8.86E-02	4.44E-02	3.19E+00	2.66E+00	8.62E-04	1.56E-04
DBO	3.68E-08	9.56E-09	2.71E-01	2.68E-01	1.72E+00	1.34E+00	7.56E-04	2.95E-04
SWO	2.87E-01	1.71E-01	2.13E+00	5.61E-01	1.52E+00	1.84E+00	8.27E-04	2.30E-03
SOA	2.57E-01	1.80E-01	1.01E+00	5.96E-01	5.71E+00	4.40E+00	2.16E-03	1.82E-03
MVO	1.74E+00	1.43E+00	7.28E-02	5.02E-02	**9.98E-01**	6.37E-12	2.77E-03	5.96E-03
IMSCSO	**3.67E-08**	**4.62E-09**	3.51E-04	2.91E-04	**9.98E-01**	4.61E-08	**3.14E-04**	**2.41E-05**

Algorithms	MF16		MF17		MF18		MF19	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	**-1.0316E+00**	4.51E-09	**3.979E-01**	1.93E-04	**3.00E+00**	4.54E-06	**-3.86E+00**	1.64E-03
WOA	**-1.0316E+00**	4.07E-12	**3.979E-01**	2.59E-07	**3.00E+00**	3.96E-06	**-3.86E+00**	1.61E-03
SCSO	**-1.0316E+00**	4.16E-11	**3.979E-01**	7.31E-09	**3.00E+00**	8.79E-07	**-3.86E+00**	2.72E-03
AOA	**-1.0316E+00**	7.34E-08	4.02E-01	3.41E-03	7.45E+00	1.01E+01	-3.85E+00	3.11E-03
HHO	**-1.0316E+00**	1.61E-12	**3.979E-01**	5.30E-08	**3.00E+00**	4.25E-09	**-3.86E+00**	1.04E-03
SCA	**-1.0316E+00**	1.38E-05	3.985E-01	9.50E-04	**3.00E+00**	9.53E-06	-3.85E+00	2.67E-03
KOA	**-1.0168E+00**	1.42E-02	4.43E-01	5.72E-02	3.55E+00	4.31E-01	-3.85E+00	6.70E-03
DO	**-1.0316E+00**	1.17E-14	**3.979E-01**	2.91E-12	**3.00E+00**	7.26E-10	**-3.86E+00**	**5.34E-09**
PSO	**-1.0316E+00**	**6.05E-16**	**3.979E-01**	**0.00E+00**	**3.00E+00**	1.62E-09	**-3.86E+00**	2.61E-15
DBO	**-1.0316E+00**	6.39E-16	**3.979E-01**	**0.00E+00**	**3.00E+00**	1.84E-09	**-3.86E+00**	3.20E-03
SWO	**-1.0316E+00**	1.26E-06	**3.979E-01**	1.77E-04	**3.00E+00**	2.45E-07	**-3.86E+00**	1.39E-04
SOA	**-1.0316E+00**	6.41E-05	4.01E-01	7.72E-03	**3.00E+00**	1.36E-05	-3.85E+00	1.82E-06
MVO	**-1.0316E+00**	7.49E-08	**3.978E-01**	1.15E-07	**3.00E+00**	1.10E-06	**-3.86E+00**	2.87E-07
IMSCSO	**-1.0316E+00**	3.88E-07	**3.979E-01**	2.29E-05	**3.00E+00**	**3.78E-10**	**-3.86E+00**	3.13E-04

Algorithms	MF20		MF21		MF22		MF23	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	-3.26E+00	7.62E-02	-9.81E+00	1.28E+00	**-10.40E+00**	**2.02E-04**	**-10.53E+00**	**1.94E-04**
WOA	-3.26E+00	6.58E-02	-9.72E+00	1.63E+00	-9.37E+00	2.32E+00	-9.25E+00	2.67E+00
SCSO	-3.25E+00	8.53E-02	-6.41E+00	2.29E+00	-6.32E+00	2.28E+00	-6.75E+00	2.51E+00
AOA	-3.13E+00	8.01E-02	-4.20E+00	1.24E+00	-4.23E+00	1.27E+00	-4.48E+00	1.48E+00
HHO	-3.19E+00	7.62E-02	-5.39E+00	1.29E+00	-5.44E+00	1.34E+00	-5.31E+00	9.85E-01
SCA	-2.96E+00	2.10E-01	-3.53E+00	1.83E+00	-4.46E+00	2.09E+00	-5.23E+00	1.61E+00
KOA	-3.07E+00	1.10E-01	-2.84E+00	1.40E+00	-2.71E+00	1.40E+00	-2.91E+00	1.40E+00
DO	-3.25E+00	6.03E-02	-6.79E+00	3.31E+00	-7.72E+00	3.40E+00	-8.45E+00	3.30E+00
PSO	-3.27E+00	5.92E-02	-8.89E+00	2.60E+00	-9.36E+00	2.41E+00	-9.74E+00	2.09E+00
DBO	-3.23E+00	1.14E-01	-7.10E+00	2.52E+00	-7.97E+00	2.86E+00	-9.10E+00	2.41E+00
SWO	-3.27E+00	5.34E-02	-8.08E+00	2.08E+00	-8.98E+00	1.71E+00	-9.40E+00	1.63E+00
SOA	-2.36E+00	6.97E-01	-7.80E+00	2.81E+00	-8.92E+00	3.60E+00	-9.11E+00	4.64E+00
MVO	-3.25E+00	6.03E-02	-7.36E+00	2.91E+00	-8.13E+00	2.88E+00	-9.10E+00	2.68E+00
IMSCSO	**-3.28E+00**	**3.20E-02**	**-9.87E+00**	**3.84E-01**	-10.13E+00	4.73E-01	-10.06E+00	6.66E-01

TABLE 5. Results of CEC 2017 functions.

Algorithms	UF17-1		MF17-3		MF17-4		MF17-5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	6.56E+07	1.48E+07	3.07E+03	2.57E+03	4.20E+02	2.12E+01	5.15E+02	6.08E+00
WOA	1.77E+07	3.40E+07	3.73E+03	2.70E+03	4.28E+02	3.38E+01	5.50E+02	2.27E+01
SCSO	1.96E+09	2.40E+09	2.12E+03	1.93E+03	4.33E+02	3.48E+01	5.33E+02	9.71E+00
AOA	7.55E+09	3.62E+09	1.01E+04	2.79E+03	1.01E+03	3.38E+02	5.54E+02	1.86E+01
HHO	5.62E+05	2.32E+05	3.09E+02	1.34E+01	4.23E+02	3.38E+01	5.45E+02	2.12E+01
SCA	8.99E+08	3.08E+08	1.87E+03	1.27E+03	4.53E+02	1.68E+01	5.48E+02	5.97E+00
KOA	7.55E+08	3.25E+08	2.11E+04	7.83E+03	5.01E+02	5.00E+01	5.69E+02	1.36E+01
DO	4.77E+03	3.80E+03	**3.00E+02**	9.34E-03	4.09E+02	1.60E+01	5.26E+02	1.05E+01
PSO	**2.85E+03**	**3.62E+03**	**3.00E+02**	**3.84E-04**	**4.03E+02**	**1.80E+00**	5.46E+02	1.33E+01
DBO	5014E+03	4.07E+03	3.23E+02	7.15E+01	4.26E+02	3.31E+01	5.35E+02	1.31E+01
SWO	1.42E+08	1.03E+08	1.27E+03	1.19E+03	4.29E+02	2.33E+01	5.38E+02	9.12E+00
SOA	8.94E+09	3.70E+09	2.17E+04	1.13E+04	1.00E+03	4.21E+02	5.92E+02	2.02E+01
MVO	7.53E+03	3.76E+03	**3.00E+02**	1.97E-02	4.06E+02	9.72E+00	5.17E+00	7.49E+00
IMSCSO	7.74E+07	1.63E+08	6.61E+02	1.71E+02	4.51E+02	2.42E+01	**5.09E+02**	**1.02E+00**

Algorithms	MF17-6		MF17-7		MF17-8		MF17-9	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	**6.00E+02**	**9.93E-01**	7.32E+02	1.10E+01	8.26E+02	5.03E+00	9.10E+02	2.00E+01
WOA	6.29E+02	1.15E+01	7.82E+02	1.95E+01	8.49E+02	1.95E+01	1.43E+03	3.77E+02
SCSO	6.13E+02	8.48E+00	7.59E+02	1.79E+01	8.27E+02	7.97E+00	1.00E+03	1.14E+02
AOA	6.37E+02	7.96E+00	7.96E+02	1.41E+01	8.31E+02	5.84E+00	1.39E+03	1.65E+02
HHO	6.30E+02	9.72E+00	7.82E+02	1.99E+01	8.29E+02	8.89E+00	1.34E+03	2.41E+02
SCA	6.18E+02	3.08E+00	7.75E+02	**1.01E+01**	8.39E+02	6.47E+00	1.01E+03	3.32E+01
KOA	6.36E+02	6.78E+00	8.27E+02	1.87E+01	8.74E+02	8.53E+00	1.61E+03	3.82E+02
DO	6.03E+02	4.00E+00	7.50E+02	1.57E+01	8.25E+02	1.17E+01	9.20E+03	4.17E+01
PSO	6.13E+02	1.05E+01	7.28E+02	6.68E+00	8.21E+02	8.04E+00	9.50E+02	1.86E+02
DBO	6.08E+02	1.03E+01	7.46E+02	1.73E+01	8.31E+02	1.14E+01	9.94E+02	1.08E+02
SWO	6.11E+02	5.06E+00	7.59E+02	1.14E+01	8.29E+02	7.44E+00	9.77E+02	8.09E+01
SOA	6.54E+02	1.70E+01	8.20E+02	2.33E+01	8.59E+02	1.48E+01	1.68E+03	3.44E+02
MVO	6.01E+02	1.94E+00	7.32E+02	1.07E+01	8.202E+02	1.07E+01	**9.00E+02**	**7.09E-01**
IMSCSO	6.30E+02	1.33E+01	**7.21E+02**	1.70E+01	**8.20E+02**	**4.43E+00**	1.10E+03	2.07E+02

Algorithms	MF17-10		HF17-11		HF17-12		HF17-13	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	1.62E+03	2.46E+02	1.14E+03	4.41E+01	6.48E+05	6.69E+05	1.16E+04	6.82E+03
WOA	2.12E+03	3.23E+02	1.19E+03	7.38E+01	3.68E+06	4.50E+06	1.44E+04	1.12E+04
SCSO	1.92E+03	3.03E+02	1.17E+03	7.12E+01	8.05E+05	1.21E+05	1.22E+04	8.56E+03
AOA	2.13E+03	2.78E+02	1.47E+03	5.28E+02	6.26E+06	1.08E+07	1.48E+04	9.28E+03
HHO	2.02E+03	3.27E+02	1.17E+03	5.06E+01	2.54E+06	2.85E+06	1.30E+04	9.35E+03
SCA	2.35E+03	2.46E+02	1.18E+03	1.97E+01	1.41E+07	1.24E+07	3.59E+04	2.79E+04
KOA	2.88E+03	2.51E+02	1.50E+03	2.64E+02	9.13E+07	5.88E+07	8.72E+05	8.01E+05
DO	1.72E+03	2.43E+02	**1.12E+03**	1.07E+01	2.66E+05	3.85E+05	1.29E+04	9.66E+03
PSO	1.96E+03	3.24E+02	1.14E+03	2.21E+01	1.26E+06	7.51E+05	7.86E+03	6.24E+03
DBO	1.95E+03	3.25E+02	1.21E+02	1.41E+02	1.83E+06	4.91E+05	1.24E+04	1.76E+04
SWO	2.38E+03	4.28E+02	1.14E+03	1.40E+01	2.73E+06	2.90E+06	1.98E+04	6.17E+04
SOA	2.79E+03	2.32E+02	4.19E+03	3.35E+03	2.15E+08	3.07E+08	4.52E+04	8.06E+04
MVO	1.72E+03	2.74E+02	1.15E+03	5.15E+01	1.42E+06	1.42E+06	1.40E+04	1.07E+04
IMSCSO	**1.53E+03**	**8.71E+01**	1.15E+03	**8.10E+00**	**1.06E+05**	**1.04E+04**	**9.70E+03**	**4.43E+03**

Algorithms	HF17-14		HF17-15		HF17-16		HF17-17	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	3.22E+03	1.90E+03	3.63E+03	1.70E+03	1.71E+03	1.19E+02	**1.75E+03**	2.69E+01
WOA	2.00E+03	9.76E+02	6.18E+03	4.36E+03	1.90E+03	1.50E+02	1.81E+03	5.87E+01
SCSO	2.60E+03	1.68E+03	3.06E+03	1.45E+03	1.78E+03	1.21E+02	1.84E+03	9.36E+01
AOA	9.71E+03	8.59E+03	1.53E+04	5.59E+03	2.05E+03	1.34E+02	1.88E+03	9.71E+01

TABLE 5. *(Continued.)* Results of CEC 2017 functions.

HHO	1.56E+03	8.00E+01	3.77E+03	1.62E+03	1.88E+03	1.27E+02	1.79E+03	5.18E+01
SCA	1.60E+03	1.04E+02	2.54E+03	9.11E+02	1.73E+03	7.15E+01	1.77E+03	**1.19E+01**
KOA	7.50E+03	9.59E+03	2.45E+04	1.64E+04	2.07E+03	1.17E+02	1.91E+03	4.25E+01
DO	1.53E+03	2.06E+02	**1.67E+03**	**3.72E+02**	**1.71E+03**	1.05E+02	1.76E+03	2.83E+01
PSO	1.97E+03	8.68E+02	3.17E+03	1.78E+03	1.86E+03	1.29E+02	1.77E+03	7.53E+01
DBO	1.73E+03	5.11E+02	4.81E+03	6.65E+03	1.76E+03	**1.01E+02**	1.77E+03	3.47E+01
SWO	1.53E+03	5.83E+01	1.77E+03	4.33E+02	1.79E+03	1.77E+02	1.76E+03	1.93E+01
SOA	1.68E+03	1.09E+02	1.02E+04	3.65E+03	2.07E+03	1.38E+02	1.87E+03	9.66E+01
MVO	1.53E+03	3.44E+01	2.95E+03	1.12E+03	1.82E+03	1.66E+02	1.80E+03	6.45E+01
IMSCSO	**1.52E+03**	**5.91E+01**	2.52E+03	1.48E+03	2.03E+03	2.05E+02	**1.75E+03**	1.45E+01
Algorithms	HF17-18		HF17-19		HF17-20		CF17-21	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	2.68E+04	**1.21E+04**	8.39E+03	6.05E+03	2.06E+03	4.89E+01	2.31E+03	**2.23E+01**
WOA	1.75E+04	1.23E+04	4.94E+04	9.86E+04	2.21E+03	6.86E+01	2.30E+03	6.26E+01
SCSO	1.86E+04	1.45E+04	5.31E+03	5.17E+03	2.09E+03	4.78E+01	2.28E+03	6.22E+01
AOA	1.62E+04	8.52E+03	3.02E+04	2.42E+04	2.16E+03	6.59E+01	2.32E+03	3.78E+01
HHO	**1.55E+04**	1.23E+04	1.01E+04	7.88E+03	2.17E+03	5.94E+01	2.32E+03	6.00E+01
SCA	1.42E+05	1.09E+05	4.21E+03	4.05E+03	2.09E+03	**2.21E+01**	2.26E+03	6.64E+01
KOA	4.63E+06	6.89E+06	4.37E+04	4.73E+04	2.23E+03	6.94E+01	2.34E+03	4.54E+01
DO	2.42E+04	1.58E+04	4.27E+03	3.33E+03	**2.05E+03**	3.88E+01	2.30E+03	5.38E+01
PSO	1.36E+04	1.05E+04	4.58E+03	2.55E+03	2.11E+03	6.13E+01	2.31E+03	6.21E+01
DBO	2.06E+04	1.54E+04	6445.342	8.14E+03	2.09E+03	5.99E+01	**2.21E+03**	3.07E+01
SWO	2.54E+04	1.69E+04	2.96E+03	3.15E+03	2.09E+03	4.31E+01	2.28E+03	6.48E+01
SOA	1.75E+05	3.62E+05	3.60E+06	8.86E+06	2.28E+03	8.31E+01	2.37E+03	4.00E+01
MVO	2.17E+04	1.43E+04	2.36E+04	7.52E+03	2.14E+03	8.67E+01	2.29E+03	4.87E+01
IMSCSO	3.13E+06	1.48E+07	**2.94E+03**	**3.09E+03**	2.20E+03	9.45E+01	2.32E+03	5.27E+01
Algorithms	CF17-22		CF17-23		CF17-24		CF17-25	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	2.31E+03	1.93E+01	2.62E+03	1.17E+01	2.74E+03	4.30E+01	2.93E+03	1.89E+01
WOA	2.32E+03	2.17E+01	2.65E+03	2.10E+01	2.78E+03	2.94E+01	2.95E+03	3.53E+01
SCSO	2.39E+03	1.20E+02	2.64E+03	1.37E+01	2.73E+03	8.80E+01	2.94E+03	2.52E+01
AOA	2.77E+03	2.50E+02	2.71E+03	3.12E+01	2.81E+03	5.84E+01	3.10E+03	7.52E+01
HHO	2.36E+03	2.53E+02	2.67E+03	3.01E+01	2.79E+03	9.26E+01	2.94E+03	2.74E+01
SCA	2.36E+03	3.32E+01	2.66E+03	1.01E+01	2.78E+03	3.43E+01	2.95E+03	**1.30E+01**
KOA	2.61E+03	1.43E+02	2.67E+03	**1.00E+01**	2.78E+03	4.68E+01	3.06E+03	3.95E+01
DO	2.32E+03	1.47E+02	2.64E+03	1.15E+01	2.77E+03	**1.89E+01**	2.92E+03	2.40E+01
PSO	2.301E+03	1.71E+01	2.69E+03	3.31E+01	2.80E+03	1.07E+02	**2.91E+03**	6.23E+01
DBO	2.307E+03	1.63E+01	2.64E+03	1.74E+01	**2.70E+03**	1.14E+02	2.93E+03	2.48E+01
SWO	2.32E+03	1.94E+01	2.66E+03	1.73E+01	2.71E+03	1.14E+02	2.93E+03	2.77E+01
SOA	3.02E+03	6.23E+02	2.70E+03	4.17E+01	2.87E+03	6.73E+01	3.49E+03	2.50E+02
MVO	2.30E+03	2.53E+01	**2.62E+03**	**5.83E+00**	2.72E+03	7.46E+01	**2.91E+03**	2.30E+01
IMSCSO	**2.30E+03**	**1.61E+01**	2.72E+03	4.03E+01	2.77E+03	1.52E+02	3.00E+03	1.40E+02
Algorithms	CF17-26		CF17-27		CF17-28		CF17-29	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	3.13E+03	3.76E+02	**3.09E+03**	2.43E+00	3.37E+03	9.40E+01	**3.19E+03**	4.56E+01
WOA	3.35E+03	4.45E+02	3.13E+03	3.71E+01	3.39E+03	1.31E+02	3.32E+03	9.29E+01
SCSO	3.03E+03	1.34E+02	3.10E+03	1.50E+01	3.35E+03	1.04E+02	3.23E+03	5.79E+01
AOA	3.82E+03	3.97E+02	3.22E+03	4.28E+01	3.70E+03	1.17E+02	3.35E+03	1.36E+02
HHO	3.42E+03	4.53E+02	3.15E+03	4.12E+01	3.35E+03	1.60E+02	3.32E+03	1.06E+02
SCA	3.07E+03	**3.31E+01**	3.10E+03	**1.65E+00**	3.29E+03	**6.94E+01**	3.23E+03	**3.31E+01**
KOA	3.39E+03	1.47E+02	3.12E+03	8.04E+00	3.45E+03	9.31E+01	3.40E+03	6.16E+01
DO	3.13E+03	3.92E+02	3.10E+03	1.08E+01	3.32E+03	1.35E+02	3.23E+03	5.70E+01
PSO	3.26E+03	4.09E+02	3.16E+03	6.18E+01	**3.14E+03**	7.87E+01	3.25E+03	6.82E+01

TABLE 5. ***(Continued.)*** **Results of CEC 2017 functions.**

DBO	3.06E+03	1.82E+02	3.10E+03	9.21E+00	3.33E+03	1.41E+02	3.23E+03	4.60E+01
SWO	3.04E+03	7.77E+01	3.13E+03	1.97E+01	3.32E+03	1.13E0+2	3.38E+03	1.86E+02
SOA	4.13E+03	4.91E+02	3.24E+03	8.52E+01	3.72E+03	2.33E+02	3.48E+03	1.46E+02
MVO	**3.02E+03**	3.50E+02	3.10E+03	2.11E+01	3.25E+03	1.48E+02	3.21E+03	5.80E+01
IMSCSO	3.35E+03	3.44E+02	3.26E+03	1.33E+02	4.01E+03	3.20E+02	3.77E+03	4.48E+02

Algorithms	CF17-30	
	Mean	SD
GWO	7.27E+05	8.25E+05
WOA	9.89E+05	1.54E+06
SCSO	7.15E+05	7.95E+05
AOA	1.10E+07	9.64E+06
HHO	1.05E+06	1.70E+06
SCA	9.39E+05	6.47E+05
KOA	3.02E+06	2.16E+06
DO	2.18E+05	4.12E+05
PSO	**2.85E+04**	**2.59E+04**
DBO	6.45E+05	7.01E+05
SWO	4.04E+06	4.42E+06
SOA	1.77E+07	1.42E+07
MVO	5.23E+05	5.67E+05
IMSCSO	1.58E+08	1.37E+08

CF20-10). Additionally, IMSCSO obtains the same mean and SD values compared to SCSO, AOA, SOA, and HHO on functions UF20-1, HF20-2, HF20-3, and HF20-4. For functions CF20-8, and CF20-9, IMSCSO obtains the same mean and SD values compared to SCSO, AOA, and HHO. For functions CF20-8, and CF20-9, IMSCSO obtains the same mean and SD values compared to SCSO, AOA, SOA, DBO, and HHO. For functions HF20-6, HF20-7, and CF20-10, the obtained results by IMSCSO are better than SCA, GWO, HHO, WOA, KOA, DO, PSO, DBO, SOA, SWO, and MVO. Hence, the results demonstrated that the IMSCSO algorithm performs well in solving CEC 2020 test problem. Additionally, it is evidence that IMSCSO belongs to the strong optimizer class by this section experiment.

G. STATISTICAL TEST ANALYSIS

To access the significant difference between the outcomes of IMSCSO and those of the other methods, the Wilcoxon rank-sum (WSR) statistical test under a significant level of 5% and the Friedman test are utilized on all test functions. The statistical comparison results of the WSR test are shown in Table 8, where the symbol '+' indicates that IMSCSO is more efficient than its competitor optimizers; '-' indicates that the competitor optimizer is more efficient than IMSCSO; and '=' indicates that the competitor optimizer's performance is similar to that of IMSCSO. All algorithms' average (AVE) ranks by Friedman test are presented in Figure 7. Table 8 demonstrates that the IMSCSO algorithm performs significantly better than its competitors. Furthermore, the IMSCSO method has the minimum AVE value (5.0347) and performed significantly better than the other algorithms (as shown in Figure 7).

H. CONVERGENCE SPEED ANALYSIS OF IMSCSO

The proposed IMSCSO's convergence curves against those of the other 13 algorithms on various test functions are shown in this section (see Figure 8). From Figure 8, it is clear that IMSCSO's performance for the unimodal test functions UF1 and UF2 is much better than other competing algorithms since it was able to obtain the lowest fitness value far faster than any of them. On the more challenging multimodal test functions (MF8, MF19, and MF21), IMSCSO could be clearly distinguished from all opponent algorithms, which could not compete with the proposed IMSCSO even after the optimization process was completed. On the hybrid and composition test functions (C19-1, HF17-4, C19-6, and C19-8), IMSCSO's convergence accuracy soon diverges from the other methods. These results demonstrate that IMSCSO is effective not just for convergence speed but also for ultimate quality. This is due to its ability to balance exploration and exploitation operators, which aids in avoiding stagnation into local minima while speeding convergence speed in the proper direction of the most promising areas acquired so far, particularly in the second half of the optimization process.

I. WALL-CLOCK TIME ANALYSIS

In this section of the experiments, IMSCSO is compared with the other 13 optimizers in the calculation of time-consuming

TABLE 6. Results of CEC 2019 functions.

Algorithms	C19-1		C19-2		C19-3		C19-4	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	6.97E+01	9.32E+01	3.28E+00	3.02E-01	1.27E+01	6.37E-04	**1.19E+01**	5.65E+00
WOA	2.90E+03	3.20E+03	1.16E+01	8.56E+00	1.12E+01	8.99E-01	5.06E+01	1.06E+01
SCSO	1.00E+00	1.33E-12	3.22E+00	1.20E-01	**1.07E+01**	2.77E-08	3.90E+01	1.35E+01
AOA	1.00E+00	0.00E+00	3.21E+00	9.36E-02	1.09E+01	5.59E-01	4.38E+01	1.54E+01
HHO	1.00E+00	2.34E-08	3.24E+00	6.05E-02	**1.07E+01**	**6.11E-10**	4.37E+01	1.57E+01
SCA	6.63E+02	9.85E+02	6.14E+00	2.39E+00	1.27E+01	4.68E-02	4.35E+01	6.26E+00
KOA	1.97E+04	7.41E+03	2.80E+01	8.35E+00	1.33E+01	4.01E-01	7.27E+01	8.52E+00
DO	7.49E+01	1.74E+02	**3.18E+00**	3.23E-01	1.27E+01	2.22E-07	2.84E+01	8.15E+00
PSO	1.28E+02	1.60E+02	3.20E+00	2.41E-01	1.27E+01	9.86E-09	4.04E+01	1.36E+01
DBO	1.35E+02	2.96E+02	3.31E+00	4.55E-01	1.11E+01	4.51E-01	3.34E+01	1.16E+01
SWO	1.02E+00	1.01E-01	3.37E+00	1.71E-01	1.16E+01	9.27E-01	3.73E+01	6.98E+00
SOA	6.79E+02	1.72E+03	6.56E+00	3.35E-01	1.16E+01	4.25E-01	8.85E+01	2.14E+01
MVO	1.43E+02	1.69E+02	3.25E+00	5.34E-01	1.27E+01	2.00E-06	2.02E+01	7.30E+00
IMSCSO	**1.00E+00**	**0.00E+00**	3.99E+00	**3.04E-02**	**1.07E+01**	8.74E-01	3.05E+01	**4.01E+00**

Algorithms	C19-5		C19-6		C19-7		C19-8	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	1.50E+00	2.30E-01	2.18E+00	8.78E-01	6.57E+02	2.32E+02	3.91E+00	5.18E-01
WOA	2.01E+00	4.24E-01	8.02E+00	1.49E+00	1.29E+03	3.86E+02	4.53E+00	3.16E-01
SCSO	2.18E+00	9.80E-01	5.66E+00	1.50E+00	9.92E+02	3.15E+02	4.09E+00	4.42E-01
AOA	4.40E+01	2.11E+01	1.03E+01	1.12E+00	1.18E+03	3.12E+02	4.67E+00	3.57E-01
HHO	2.01E+00	3.17E-01	7.26E+00	1.53E+00	1.15E+03	3.47E+02	4.56E+00	3.13E-01
SCA	7.56E+00	1.59E+00	6.91E+00	1.12E+00	1.45E+03	1.29E+02	4.32E+00	2.13E-01
KOA	2.05E+01	7.31E+00	1.11E+01	9.77E-01	2.03E+03	2.78E+02	5.05E+00	2.92E-01
DO	1.27E+00	1.28E-01	4.68E+00	1.27E+00	8.33E+02	2.74E+02	4.07E+00	3.56E-01
PSO	1.30E+00	1.87E-01	1.28E+02	1.60E+02	**3.25E+00**	1.28E+00	3.99E+00	4.51E-01
DBO	**1.19E+00**	1.37E-01	1.35E+02	2.96E+02	6.17E+00	1.65E+00	4.17E+00	4.41E-01
SWO	3.51E+00	1.10E+00	**1.02E+00**	5.01E-01	5.65E+00	1.41E+00	4.16E+00	3.58E-01
SOA	9.05E+01	2.97E+01	6.79E+02	1.72E+03	1.07E+01	**1.13E+00**	4.89E+00	2.64E-01
MVO	1.32E+00	**1.11E-01**	1.43E+02	1.69E+02	3.86E+00	1.67E+00	3.91E+00	3.46E-01
IMSCSO	4.66E+00	8.01E-01	5.57E+00	**4.96E-01**	1.69E+03	7.34E+02	**3.58E+00**	**2.12E-01**

Algorithms	C19-9		C19-10	
	Mean	**SD**	**Mean**	**SD**
GWO	**1.15E+00**	**6.28E-02**	2.14E+01	8.25E-02
WOA	1.37E+00	1.85E-01	2.12E+01	1.46E-01
SCSO	1.28E+00	1.33E-01	2.11E+01	7.06E-02
AOA	2.04E+00	6.02E-01	2.11E+01	5.32E-02
HHO	1.36E+00	1.44E-01	2.11E+01	6.34E-02
SCA	1.51E+00	1.21E-01	2.14E+01	8.52E-02
KOA	2.24E+00	2.86E-01	2.17E+01	1.14E-01
DO	1.22E+00	1.16E-01	2.10E+01	**3.17E-02**
PSO	1.18E+00	9.02E-02	2.16E+01	3.70E+00
DBO	1.35E+00	1.57E-01	**2.09E+01**	2.25E+00
SWO	1.35E+00	1.01E-01	2.12E+01	1.83E+00
SOA	3.44E+00	9.75E-01	2.16E+01	7.52E-02
MVO	1.21E+00	7.10E-02	2.11E+01	5.80E-02
IMSCSO	1.37E+00	8.17E-02	2.12E+01	1.27E+00

tests in the 72 test functions (i.e., 23 classical test functions, CEC2017, CEC2019, and CEC2020) mentioned above. The time-consuming calculation is that all participants run each function independently 30 times and record the time percentage results in Figures 9 to 12. As can be observed from the data in these figures, the computation of IMSCSO takes relatively more extended time, because the HOBL, DRST, and JOS mechanisms require more computing power. In general, even though it takes a lot of time, IMSCSO still outperforms other algorithms in terms of efficacy, therefore the time results are to be expected.

VI. THE PERFORMANCE OF IMSCSO ON CLASSICAL ENGINEERING PROBLEMS

This section considers seven well-known practical engineering challenges (namely the three-bar truss (TBR) case, the tension/compression spring (TCS) case, the cantilever beam (CB) case, the pressure vessel (PV) case, the speed

TABLE 7. Results of CEC 2020 functions.

Algorithms	UF20-1		MF20-2		HF20-3		HF20-4	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	4.50E+04	9.24E+04	1.34E+03	2.93E+02	7.29E+02	1.71E+01	1.901E+03	6.31E-01
WOA	4.51E+04	1.79E+05	1.79E+03	5.18E+02	7.14E+02	1.60E+01	1.902E+03	1.79E+00
SCSO	**1.00E+02**	**0.00E+00**	**1.10E+03**	**0.00E+00**	**7.00E+02**	**0.00E+00**	**1.90E+03**	**0.00E+00**
AOA	**1.00E+02**	**0.00E+00**	**1.10E+03**	**0.00E+00**	**7.00E+02**	**0.00E+00**	**1.90E+03**	**0.00E+00**
HHO	**1.00E+02**	**0.00E+00**	**1.10E+03**	**0.00E+00**	**7.00E+02**	**0.00E+00**	**1.90E+03**	**0.00E+00**
SCA	7.77E+05	1.05E+06	1.84E+03	5.59E+02	7.11E+02	1.86E+01	1.901E+03	1.67E+00
KOA	1.47E+09	3.64E+08	2.94E+03	1.64E+02	8.29E+02	2.30E+01	2.02E+03	1.54E+02
DO	2.89E+03	6.37E+03	1.59E+03	2.83E+02	7.24E+02	2.32E+01	1.901E+03	9.91E-01
PSO	1.97E+03	2.43E+03	1.72E+03	2.28E+02	7.23E+02	7.74E+00	1.901E+03	6.29E-01
DBO	2.72E+03	1.03E+04	1.55E+03	5.47E+02	7.18E+02	2.39E+01	1.901E+03	9.74E-01
SWO	2.01E+02	2.99E+02	1.15E+03	2.49E+02	7.01E+02	3.10E+00	1.90E+03	2.73E-01
SOA	**1.00E+02**	**0.00E+00**	**1.10E+03**	**0.00E+00**	**7.00E+02**	**0.00E+00**	**1.90E+03**	**0.00E+00**
MVO	1.42E+04	9.46E+03	1.79E+03	2.91E+02	7.30E+02	8.24E+00	1901.623	0.793173
IMSCSO	**1.00E+02**	**0.00E+00**	**1.10E+03**	**0.00E+00**	**7.00E+02**	**0.00E+00**	**1.90E+03**	**0.00E+00**
Algorithms	HF20-5		HF20-6		HF20-7		CF20-8	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GWO	2.76E+03	2.14E+03	1.61E+03	2.42E+03	2.11E+03	2.01E+01	2.25E+03	1.12E+02
WOA	3.27E+04	5.86E+04	1.65E+03	8.36E+01	1.01E+05	2.01E+05	2.73E+03	5.42E+02
SCSO	**1.70E+03**	**0.00E+00**	1.60E+03	8.44E-06	2.10E+03	4.09E-08	**2.20E+03**	**0.00E+00**
AOA	**1.70E+03**	**0.00E+00**	1.60E+03	4.22E-14	**2.10E+03**	**0.00E+00**	**2.20E+03**	**0.00E+00**
HHO	**1.70E+03**	**0.00E+00**	1.60E+03	6.69E-05	2.10E+03	1.14E-06	**2.20E+03**	**0.00E+00**
SCA	1.06E+04	1.54E+04	1.62E+03	2.70E+01	2.13E+03	6.90E+01	2.53E+03	5.31E+02
KOA	4.62E+05	3.73E+05	1.88E+03	9.98E+01	1.72E+05	2.77E+05	3.72E+03	2.57E+02
DO	3.71E+03	3.50E+03	1.62E+03	3.92E+01	2.25E+03	1.66E+02	2.66E+03	4.42E+02
PSO	5.06E+03	3.84E+03	1.62E+03	3.52E+01	2.66E+03	4.30E+02	2.42E+03	1.48E+02
DBO	6.07E+03	6.24E+03	1.61E+03	2.88E+01	2.19E+03	1.78E+02	**2.20E+03**	**0.00E+00**
SWO	1.83E+03	2.28E+02	1.60E+03	3.75E-01	2.102E+03	7.07E+00	2.20E+03	2.38E-03
SOA	3.42E+05	9.91E+05	1.60E+03	3.08E-04	5.81E+05	8.01E+05	2.24E+03	1.34E+02
MVO	6.66E+03	3.90E+03	1.69E+03	8.19E+01	5.19E+03	1.88E+03	2.60E+03	4.20E+02
IMSCSO	**1.70E+03**	**0.00E+00**	**1.60E+03**	**0.00E+00**	**2.10E+03**	**0.00E+00**	**2.20E+03**	**0.00E+00**
Algorithms	CF20-9		CF20-10					
	Mean	SD	Mean	SD				
GWO	2.40E+03	1.19E-13	2.57E+03	5.65E+00				
WOA	2.40E+03	1.91E+00	2.59E+03	1.22E+01				
SCSO	**2.40E+03**	**0.00E+00**	2.50E+03	7.73E-05				
AOA	**2.40E+03**	**0.00E+00**	**2.50E+03**	**0.00E+00**				
HHO	**2.40E+03**	**0.00E+00**	2.50E+03	1.05E-04				
SCA	2.40E+03	3.68E-13	2.58E+03	1.53E+01				
KOA	2.53E+03	2.96E+01	2.62E+03	1.90E+01				
DO	2.401E+03	2.40E+00	2.57E+03	1.68E+01				
PSO	2.40E+03	1.26E+00	2.57E+03	1.25E+01				
DBO	**2.40E+03**	**0.00E+00**	2.55E+03	3.83E+01				
SWO	2.40E+03	7.77E-02	2.51E+03	4.63E+00				
SOA	**2.40E+03**	**0.00E+00**	2.58E+03	6.68E+01				
MVO	2.40E+03	8.97E-02	2.57E+03	1.05E+01				
IMSCSO	**2.40E+03**	**0.00E+00**	**2.50E+03**	**0.00E+00**				

reducer (SR) case, the I-beam vertical deflection (IBVD) case, and the piston lever (PL) case) to further demonstrate the applicability of IMSCSO in dealing with practical engineering problems. Numerous non-linear and complex constraints that are based on design criteria, resource limitations, and security needs are frequently present in

TABLE 8. Statistical results of Wilcoxon rank-sum test of IMSCSO.

Functions	IMSCSO vs. GWO (+/=/-)	IMSCSO vs. WOA (+/=/-)	IMSCSO vs. SCSO (+/=/-)	IMSCSO vs. AOA (+/=/-)	IMSCSO vs. HHO (+/=/-)	IMSCSO vs. SCA (+/=/-)	IMSCSO vs. KOA (+/=/-)	IMSCSO vs. DO (+/=/-)
23 classical test functions	18/0/5	17/0/6	18/3/2	19/3/1	19/3/0	23/0/0	23/0/0	20/0/3
CEC 2017	25/0/4	19/0/10	23/0/6	19/0/10	20/0/9	22/0/7	26/0/3	27/0/2
CEC 2019	9/0/1	10/0/0	9/0/1	9/1/0	9/0/1	5/1/4	10/0/0	9/0/1
CEC 2020	10/0/0	10/0/0	2/7/1	0/9/1	3/7/0	10/0/0	10/0/0	10/0/0
Total	**62**/0/10	**56**/0/16	**52**/10/10	**47**/13/12	**52**/10/10	**61**/1/11	**69**/0/3	**66**/0/6

Functions	IMSCSO vs. PSO (+/=/-)	IMSCSO vs. DBO (+/=/-)	IMSCSO vs. SWO (+/=/-)	IMSCSO vs. SOA (+/=/-)	IMSCSO vs. MVO (+/=/-)
23 classical test functions	22/0/1	16/1/6	19/0/4	19/1/3	16/3/4
CEC 2017	23/0/6	25/0/4	27/0/2	25/0/4	25/0/4
CEC 2019	9/0/1	6/0/4	8/0/2	9/0/1	10/0/0
CEC 2020	10/0/0	8/2/0	10/0/0	4/5/1	10/0/0
Total	**64**/0/8	**55**/3/14	**64**/0/8	**57**/6/9	**61**/3/8

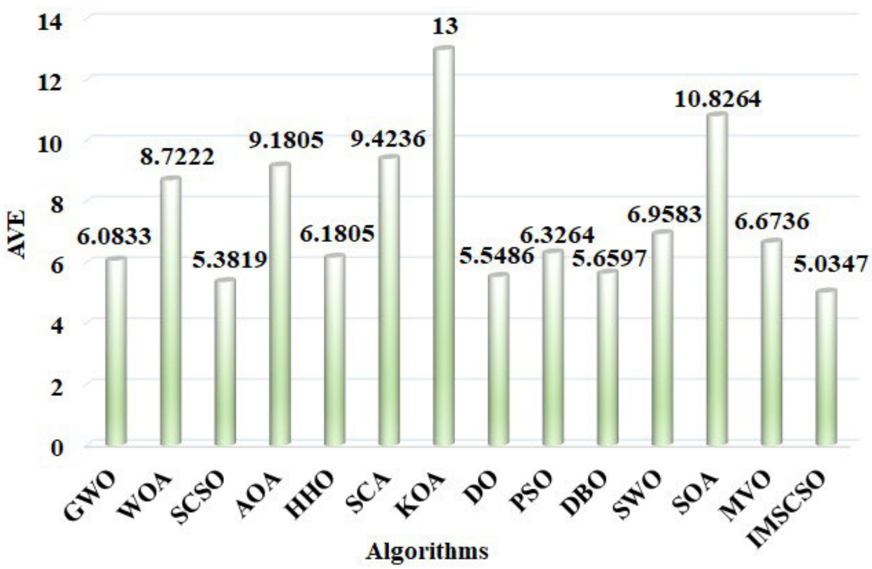

FIGURE 7. The AVE ranks by Friedman test for all algorithms.

these issues. Additionally, a method for simultaneously handling constraints and objective functions is to add a penalty component to the objective functions, which will turn these problems into unconstrained optimization problems [88].

A. TBR CASE

The TBR case is a typical engineering optimization problem. Figure 13 shows the components of the TBR (W_1 (= c_1), W_2 (= c_2), and W_3 (= c_3)) under the restrictions of buckling, stress, and deflection. The objective of this case is to obtain the lightest weight with the optimal values of the two bars ($W_1 = W_3$, W_2). Additionally, the mathematical formulation of the TBR situation is provided by Eq. (22).

$$\begin{cases} F_{\min}(C) = 100 \times (2\sqrt{2}\, c_1 + c_2) \\ S.t.\, g_1(C) = 2\dfrac{\sqrt{2}\, c_1 + c_2}{\sqrt{2}c_1^2 + 2c_1c_2} - 2 \leq 0 \\ g_2(C) = 2\dfrac{c_2}{\sqrt{2}\, c_1^2 + 2c_1c_2} - 2 \leq 0 \\ g_3(C) = \dfrac{1}{\sqrt{2}\, c_2 + c_1} - 2 \leq 0 \\ 0 \leq c_1, c_2 \leq 1 \end{cases} \quad (22)$$

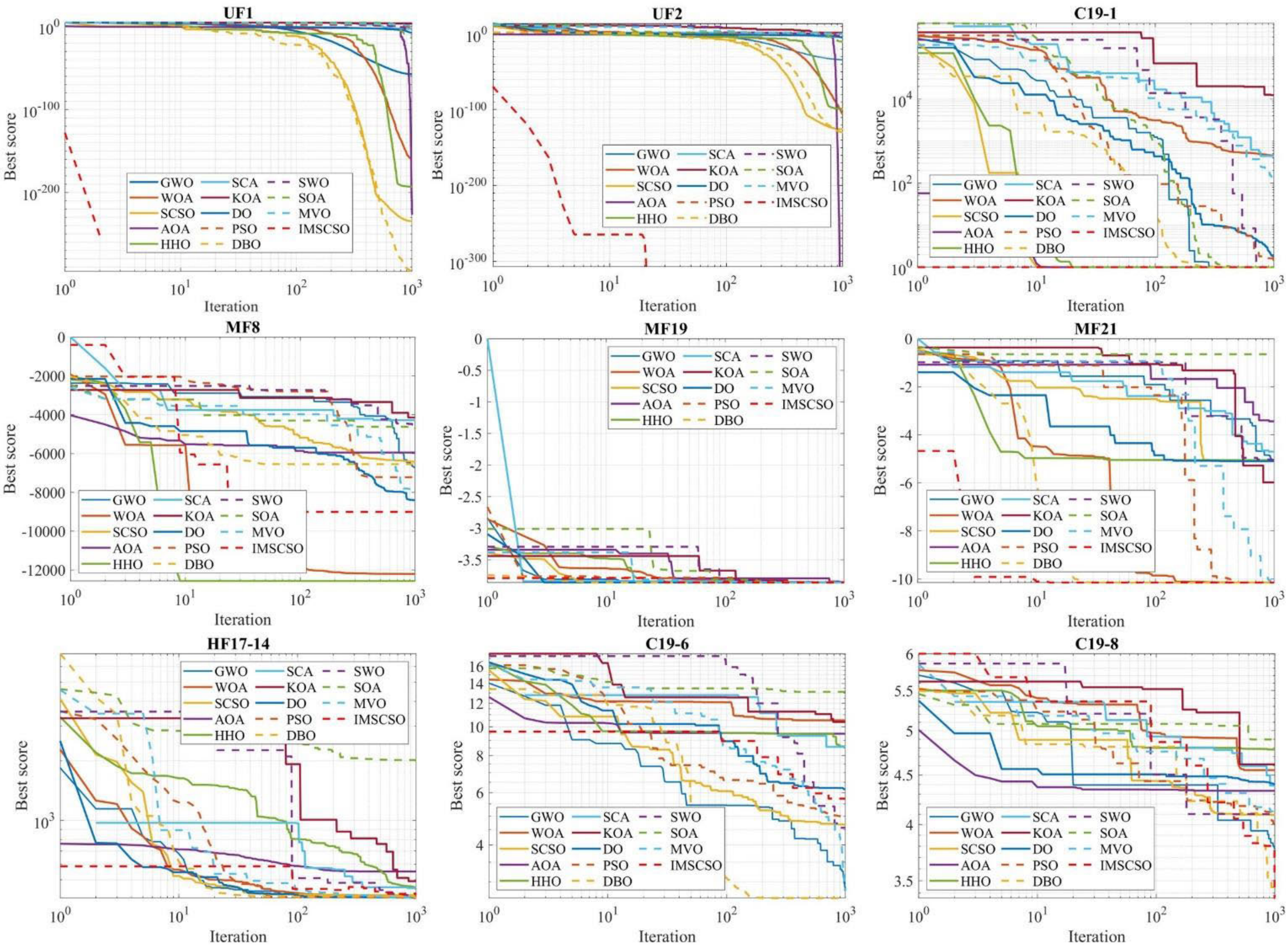

FIGURE 8. Convergence curves of the IMSCSO and other 8 optimizers algorithm on some test functions.

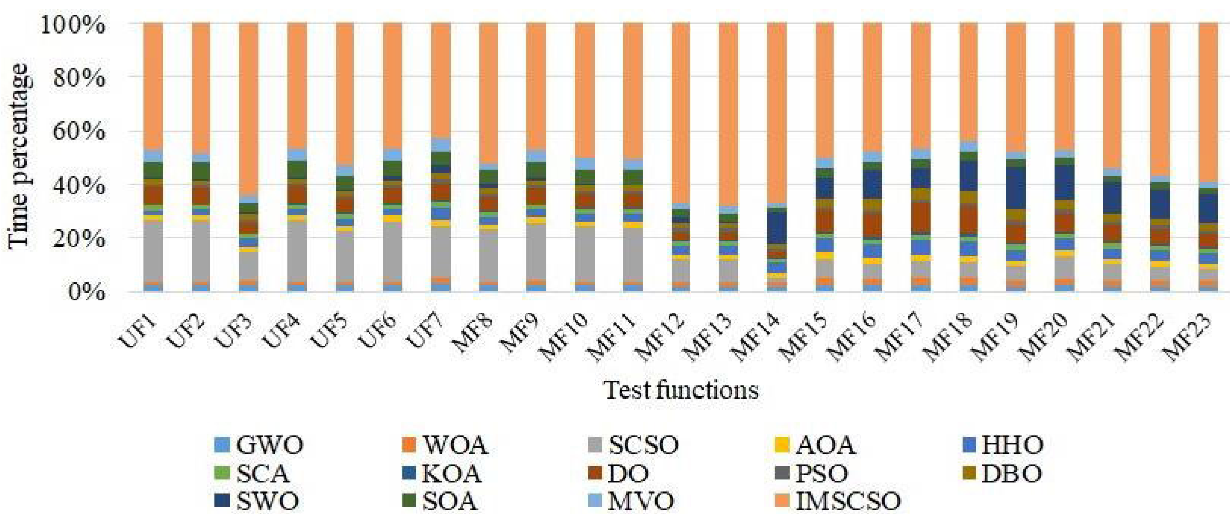

FIGURE 9. Time percentage results of 23 classical test functions.

Numerous metaheuristic algorithms, including MFO [25], YDSE [37], INFO [54], CS [89], ALO [90], MBA [91], and DEDS [92], were used to optimize this situation. Table 9 demonstrated that the IMSCSO achieved a very promising result with the best objective function. The collected findings again demonstrated that IMSCSO is capable of handling difficult constraint situations effectively.

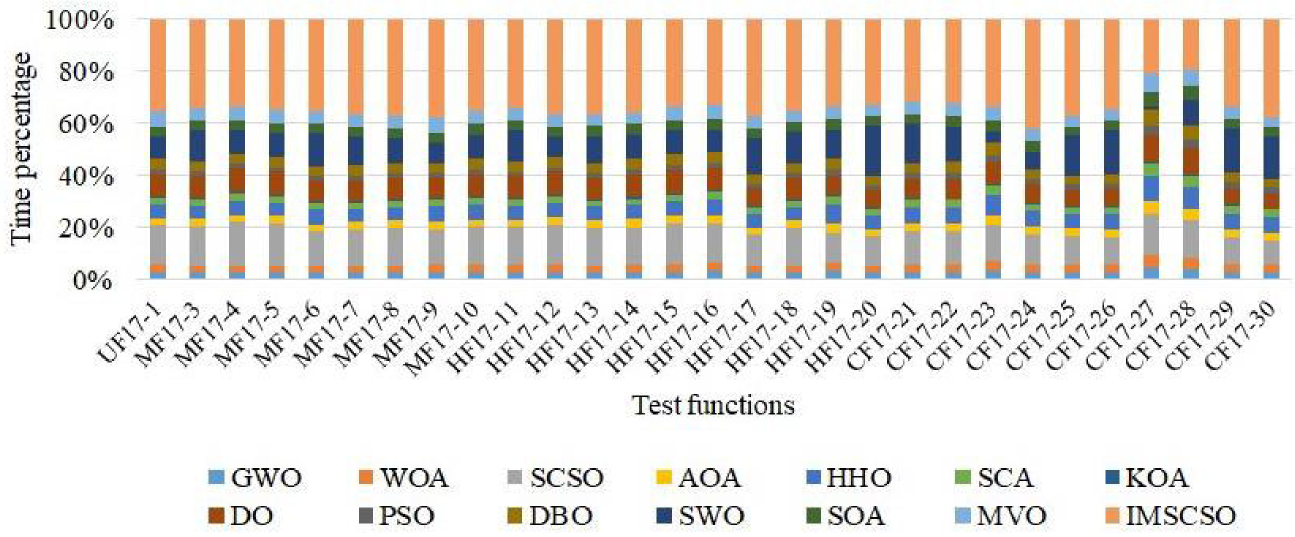

FIGURE 10. Time percentage results of CEC 2017 test functions.

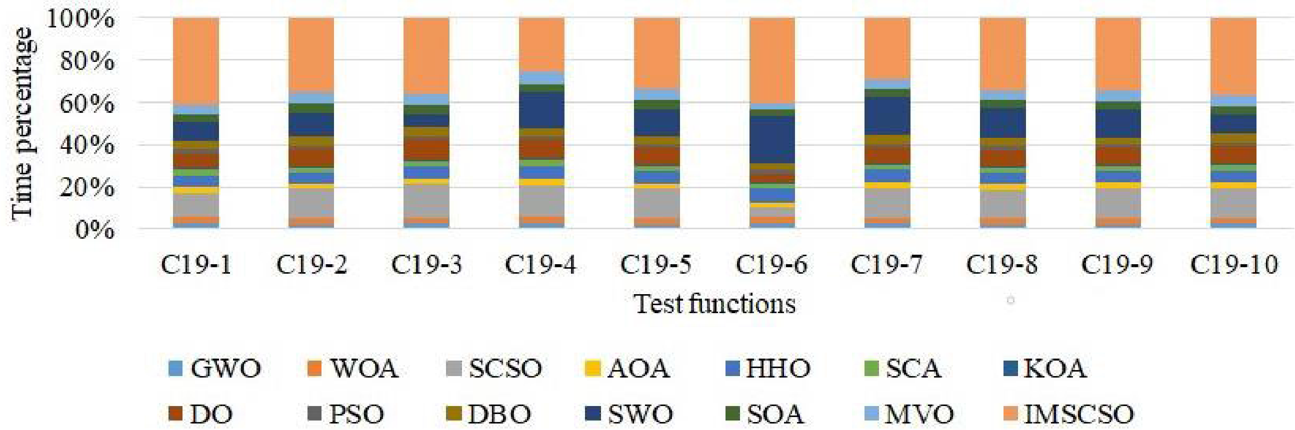

FIGURE 11. Time percentage results of CEC 2019 test functions.

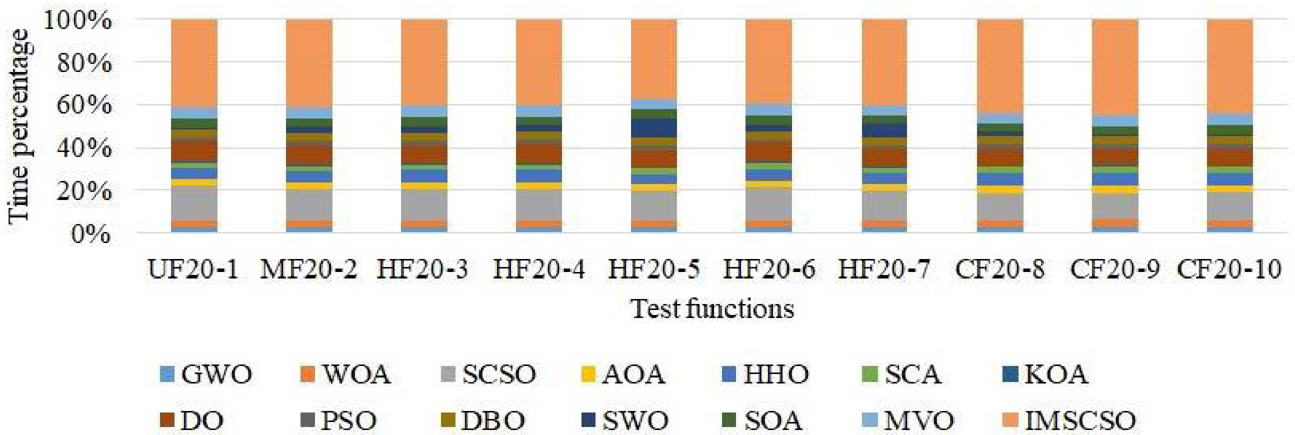

FIGURE 12. Time percentage results of CEC 2020 test functions.

B. TCS CASE

Figure 14 depicts the TCS's organizational structure. The goal of this instance is to decrease the weight of TCS by optimizing the three parameters (i.e., $D(= c_1)$, $W(= c_2)$, $LS(= c_3)$) while taking into account the restrictions of shear stress, minimum deflection, surge frequency, and limits on

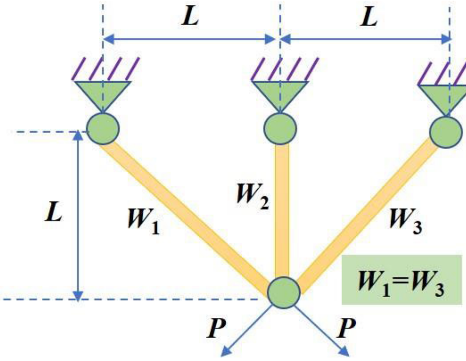

FIGURE 13. Components of the TBR problem.

TABLE 9. Results of IMSCSO and other algorithms for the TBR problem.

Methods	Optimal variables		Optimal value
	c_1	c_2	
MBA [91]	0.788565	0.4085597	263.8958522
DEDS[92]	0.78867513	0.40824828	263.8958434
CS [89]	0.788670	0.409020	263.97160
ALO [90]	0.78866280	0.4082831330	263.8958434
MFO [25]	0.788244770	0.409466905	263.8959796
INFO [54]	0.788672734	0.408255081	263.8958434
YDSE[37]	0.78868	0.40825	263.90
IMSCSO	0.788674036299621	0.408251396540540	263.895843339345

the outside diameter. The TCS situation is mathematically expressed as follows in Eq. (23). In contrast to previous metaheuristic algorithms, Table 10 clearly indicates that the IMSCSO algorithm performs better but is similar to the YDSE algorithm.

$$\begin{cases} F_{\min}(C) = (2+c_3)c_2c_1^2 \\ S.t.\, g_1(C) = 1 - \frac{c_2^3 c_3}{71785 c_1^4} \leq 0 \\ g_2(C) = \frac{4c_2^2 - c_1c_2}{12566(c_2c_1^3 - c_1^4)} + \frac{1}{5108c_1^2} - 1 \leq 0 \\ g_3(C) = 1 - \frac{140.45c_1}{c_2^2 c_3} \leq 0 \\ g_4(C) = \frac{c_1+c_2}{1.5} - 1 \leq 0 \\ 0.05 \leq c_1 \leq 2,\ 0.25 \leq c_2 \leq 1.3,\ 2 \leq c_3 \leq 15 \end{cases} \tag{23}$$

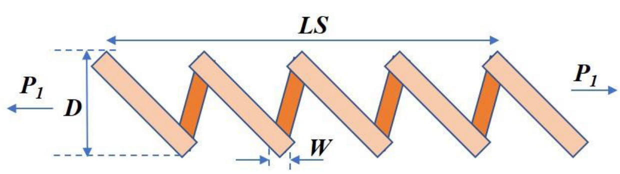

FIGURE 14. The structure of the TCS.

C. CB CASE

As illustrated in Figure 15, this CB typically comprises five hollow square pieces, each of the same thickness and boosted from the first one. While the fifth section of the beam is subject to a vertical force, the goal is to reduce the weight of the beam. Five decision variables (c_1, c_2, c_3, c_4, and c_5) with a single constraint are represented by the lengths of the five pieces. The equation is written as Eq. (24). By observing the results in Table 11, compared to other algorithms, the IMSCSO shows a good performance in solving the CB problem, with a best beam weight of 1.33995.

$$\begin{cases} F_{\min}(C) = 0.0624(c_1+c_2+c_3+c_4+c_5) \\ S.t.\, g(C) = \frac{61}{c_1^3} + \frac{37}{c_2^3} + \frac{19}{c_3^3} + \frac{7}{c_4^3} + \frac{1}{c_5^3} - 1 \leq 0 \\ 0.1 \leq c_1, c_2, c_3, c_4, c_5 \leq 100 \end{cases} \tag{24}$$

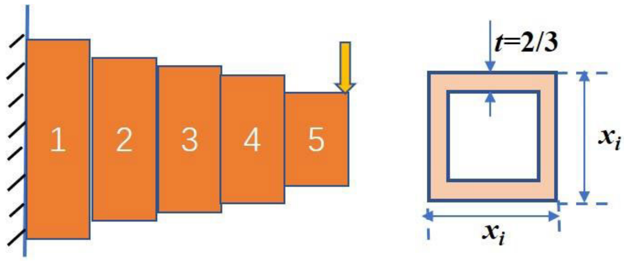

FIGURE 15. The structure of the TCS.

D. PV CASE

By choosing the best values for four design variables (i.e., the shell thickness ($TS = c_1$), the head thickness ($TH = c_2$), the inner radius ($R = c_3$), and the length of the cylindrical component ($l = c_4$)) while using pressure needs as an optimization constraint, the PV example seeks to lower overall output. Figure 16 shows the structure of the PV. This case's mathematical model is shown in Eq. (25). The simulation results in Table 12 show that the IMSCSO method beats other commonly used algorithms (except for KOA algorithm) in terms of determining the optimum cost.

$$\begin{cases} F_{\min}(C) = 0.6224c_1c_3c_4 + 1.7781c_2c_3^2 + 3.1661c_1^2c_4 \\ +19.84c_1^2c_3 \\ S.t.\, g_1(C) = -c_1 + 0.0193c_3 \leq 0 \\ g_2(C) = -c_2 + 0.00954c_3 \leq 0 \\ g_3(C) = -\pi c_3^2 c_4 - \frac{4\pi}{3}c_3^3 + 1296000 \leq 0 \\ g_4(C) = c_4 - 240 \leq 0 \\ c_1,\ c_2 \in \{1 \times 0.0625,\ 2 \times 0.0625, \cdots,\ 1600 \times 0.0625\}, \\ 10 \leq c_3, c_4 \leq 200 \end{cases} \tag{25}$$

E. SR CASE

This issue (as seen Figure 17) is a well-known mechanical system design issue. The SR is one of the critical components

TABLE 10. Results of IMSCSO and other algorithms for the TCS problem.

Methods	Optimal variables			Optimal value
	c_1	c_2	c_3	
WOA [22]	0.0512070	0.3452150	12.004030	0.012676
ES [94]	0.51989	0.363965	10.89052	0.012681
HS [93]	0.0511540	0.3498710	12.076430	0.012678
SCSO [30]	0.0500	0.3175	14.0200	0.01271702
KOA [38]	0.051689	0.356715	11.289151	0.0126652328
INFO [54]	0.051555	0.353499	11.48034	0.012666
YDSE [37]	0.051689	0.35673	11.288	0.012665
IMSCSO	0.051671	0.356307	11.31307	0.012665

TABLE 11. Results of IMSCSO and other algorithms for the CB problem.

Methods	Optimal variables					Optimal value
	c_1	c_2	c_3	c_4	c_5	
ALO [90]	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
CS [89]	6.0089	5.3049	4.5023	3.5077	2.1504	1.3399
SOS [95]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
SCSO [30]	6.0164	5.3060	4.4935	3.5059	2.1516	1.3399524
KOA [38]	6.0160	5.3092	4.4943	3.5015	2.1527	1.339956
SMA [27]	6.017757	5.310892	4.493758	3.501106	2.150159	1.33996
YDSE [37]	6.0160	5.3092	4.4943	3.5015	2.1527	1.3400
IMSCSO	6.01535	5.30936	4.49713	3.49850	2.15332	1.33995

TABLE 12. Results of IMSCSO and other algorithms for the PV problem.

Methods	Optimal variables				Optimal value
	c_1	c_2	c_3	c_4	
KOA [38]	0.778179	0.384659	40.319619	200	5885.434175
YDSE [37]	1.3233	7.0746	42.098	176.64	6059.7
WOA [22]	0.8125	0.4375	42.0982699	176.638998	6059.7410
SCSO [30]	0.7798	0.9390	40.3864	199.2918	5917.46
GWO [21]	0.8125	0.4345	42.0892	176.7587	6051.5639
AO [96]	1.0540	0.182806	59.6219	38.8050	5949.2258
ES [94]	0.8125	0.4375	42.098087	176.640518	6059.74560
MVO [41]	0.8125	0.4375	42.090738	176.73869	6060.8066
SMA [27]	0.7931	0.3932	40.6711	196.2178	5994.1857
IMSCSO	0.7782088	0.3846886	40.32179	199.976	5885.60827

of the gearbox in this situation and has a variety of uses. The weight of the SR is dependent on 11 limitations in this optimization question, all of which must be minimized [67]. The remaining nine are inequalities with linear constraints, with seven of them being nonlinear constraints. These four factors are the surface stress, stresses in the shafts, transverse shaft deflections, and bending stress of the gear teeth. Additionally, there are seven variables in this problem: face width $RW(c_1)$, module of teeth $RM(c_2)$, the number of teeth in the pinion $RN(c_3)$, length of the first shaft between bearings $RL_2(c_4)$, length of the second shaft between bearings $RL_1(c_5)$, the diameter of first shafts $RD_1(c_7)$, and the diameter of second shafts $RD_2(c_6)$. The equation of the SR case is given in Eq. (26). In comparison to other metaheuristic methods, the findings obtained, which are given in Table 13, demonstrate that the IMSCSO method finds a least-cost design value and

TABLE 13. Results of IMSCSO and other algorithms for the SR problem.

Optimal variables	Methods						
	SCA [50]	YDSE [37]	HS [93]	AO [96]	GSA [33]	CS [89]	IMSCSO
c_1	3.508755	3.50	3.520124	3.5021	3.6	3.5015	3.50
c_2	0.7	0.7	0.7	0.7	0.7	0.7	0.7
c_3	17	17	17	17	17	17	17
c_4	7.3	7.3	8.37	7.3099	8.3	7.605	7.300
c_5	7.8	7.7153	7.8	7.7476	7.8	7.8181	7.71532
c_6	3.461020	3.3505	3.366970	3.3641	3.369658	3.352	3.35054
c_7	5.289213	5.2867	5.288719	5.2994	5.289224	5.2875	5.28665
Optimal value	3030.563	2994.4	3029.002	3007.7328	3051.12	3000.981	2994.42

TABLE 14. Results of IMSCSO and other algorithms for the PL problem.

Methods	Optimal variables				Optimal value
	c_1	c_2	c_3	c_4	
YDSE [37]	0.05	2.0415	4.0830	120	8.4127
CS [89]	0.05	2.043	4.085	120	8.427
SCSO [30]	0.05	2.04	4.083	119.99	8.40901438899551
SNS [88]	0.050	2.042	4.083	120	8.412698349
HPSO [97]	135.5	2.48	4.75	116.62	162
IMSCSO	0.05	2.041513	4.083027	120.00	8.412698323205378

closely follows the YDSE algorithm.

$$\begin{cases} F_{\min}(C) = 0.7854c_1c_2^2(3.3333c_3^2 + 14.9334c_3 - 43.0934) \\ -1.508c_1(c_6^2 + c_7^2) + 7.4777(c_6^3 + c_7^3) + 0.7854 \\ (c_4c_6^2 + c_5c_7^2) \\ S.t.\ g_1(C) = \dfrac{27}{c_1c_2^2c_3} - 1 \le 0 \\ g_2(C) = \dfrac{397.5}{c_1c_2^2c_3^2} - 1 \le 0 \\ g_3(C) = \dfrac{1.93c_5^3}{c_2c_6^4c_3} - 1 \le 0 \\ g_4(C) = \dfrac{\sqrt{(745c_4/c_2c_3)^2 + 16.9 \times 10^6}}{110c_6^3} - 1 \le 0 \\ g_5(C) = \dfrac{\sqrt{(745c_5/c_2c_3)^2 + 157.5 \times 10^6}}{85c_7^3} - 1 \le 0 \\ g_6(C) = \dfrac{c_2c_3}{40} - 1 \le 0 \\ g_7(C) = \dfrac{5c_2}{c_1} - 1 \le 0 \\ g_8(C) = \dfrac{1.93c_5^3}{c_2c_7^4c_3} - 1 \le 0 \\ g_9(C) = \dfrac{c_1}{12c_2} - 1 \le 0 \\ g_{10}(C) = \dfrac{1.5c_6 + 1.9}{c_4} - 1 \le 0 \\ g_{11}(C) = \dfrac{1.1c_7 + 1.9}{c_5} - 1 \le 0 \\ 2.6 \le c_1 \le 3.6,\ 0.7 \le c_2 \le 0.8,\ 7.3 \le c_4, c_5 \le 8.3, \\ 2.9 \le c_6 \le 3.9, \\ 5 \le c_7 \le 5.5,\ c_3 \in \{17, 18, \cdots, 28\} \end{cases} \tag{26}$$

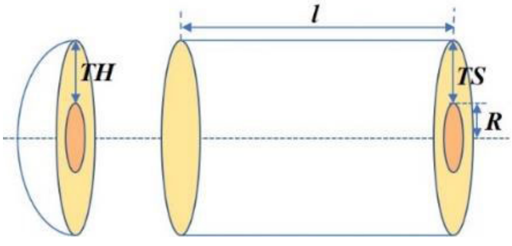

FIGURE 16. The structure of the PV.

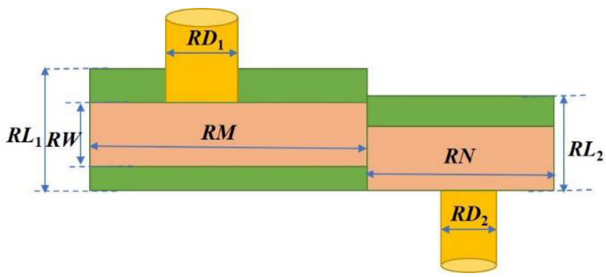

FIGURE 17. The structure of the SR.

F. PL CASE

For the PL case, as shown in Figure 18, we are attempting to decrease the oil volume while raising the piston lever from 0 to 45 to identify the four piston elements, $H(c_1)$, $B(c_2)$, $D(c_3)$, and $X(c_4)$. The mathematical equations can be expressed on Eq. (27). Additionally, Table 14 compiles an analysis of the algorithms. Simulated findings show that IMSCSO beats its competitors in determining the minimum volume of the oil and is less than SCSO algorithm.

TABLE 15. Results of IMSCSO and other algorithms for the IBVD problem.

Methods	Optimal variables				Optimal value
	c_1	c_2	c_3	c_4	
INFO [54]	80.00	50.00	0.90	2.32	0.0130741
CS [89]	80.00	50.00	0.90	2.32	0.0130747
ARSM [98]	80.00	37.05	1.71	2.31	0.0157
IARSM [98]	79.99	48.42	0.90	2.40	0.131
SOS [95]	80.00	50.00	0.90	2.32	0.0130741
IMSCSO	80.00	50.00	0.90	2.32179	0.0130741

G. IBVD CASE

The IBVD case was used to further validate IMSCSO's capabilities. The IBVD problem seeks to reduce the vertical deflection of the I-beam shown in Figure 19 as much as possible. The four variables are the length (c_1), height (c_2), thicknesses of the beam web (c_4), and flange (c_3). The problem formulation was defined in Eq. (28). The IBVD case was optimized by ARSM [98], IARSM [98], SOS [95], INFO [54], and CS [89]. According to Table 15, which summarizes the outcomes of all optimizers, IMSCSO outperforms CS, ARSM, and IARSM in terms of minimizing vertical deflection and is similar to INFO and SOS.

$$\begin{cases} F_{\min}(C)=\frac{\pi}{4}c_3^2\left(\begin{array}{l}\sqrt{(c_4\sin 45^\circ+c_1)^2+(c_2-c_4\cos 45^\circ)^2}\\ -\sqrt{(c_4-c_2)^2+c_1^2}\end{array}\right)\\ S.t.\ g_1(C)=10000\times 240\times\cos 45^\circ-\\ \frac{1500\pi c_3^2\left|-c_4(c_4\sin 45^\circ+c_1)+c_1(c_2-c_4\cos 45^\circ)\right|}{4\sqrt{(c_4-c_2)^2+c_1^2}}\leq 0\\ g_2(C)=10000\times(240-c_4)-1.8\times 10^6\leq 0\\ g_3(C)=1.2\left(\begin{array}{l}\sqrt{(c_4\sin 45^\circ+c_1)^2+(c_2-c_4\cos 45^\circ)^2}\\ -\sqrt{(c_4-c_2)^2+c_1^2}\end{array}\right)\\ -\sqrt{(c_4-c_2)^2+c_1^2}\leq 0\\ g_4(C)=\frac{c_3}{2}-c_2\leq 0\\ 0.05\leq c_1,\ c_2,c_4\ \leq 500,\ \ 0.05\leq c_3\leq 120 \end{cases} \tag{27}$$

$$\begin{cases} F_{\min}(C)=\frac{5000}{c_3(c_2-2c_4)^3/12+(c_1c_4^3/6)+2c_1c_4(c_2-c_4/2)^2}\\ S.t.\ g_1(C)=2c_1c_3+c_3(c_2-2c_4)\leq 300\\ g_2(C)=\frac{18\times 10^4c_2}{c_3(c_2-2c_4)^3+2c_1c_3(4c_4^2+3c_2(c_2-2c_4))}\\ +\frac{15\times 10^3c_1}{(c_2-2c_4)c_3^2+2c_3c_1^3}\leq 56\\ 10\leq c_1\leq 50,\ 10\leq c_2\leq 80,\ 0.9\leq c_3,c_4\leq 5 \end{cases} \tag{28}$$

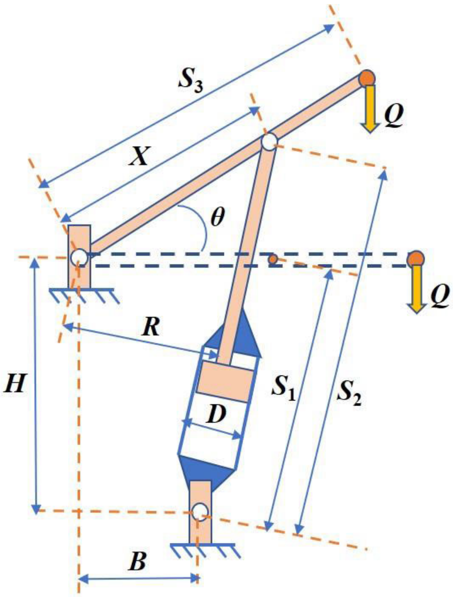

FIGURE 18. The structure of the PL.

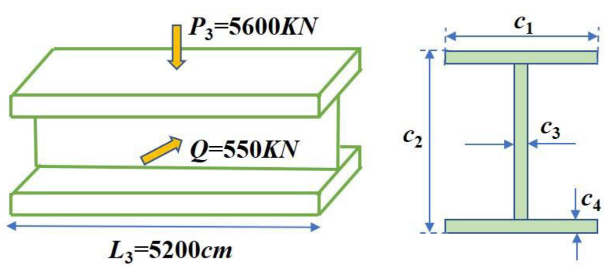

FIGURE 19. The structure of the IBVD.

VII. CONCLUSION AND FUTURE WORK

In this paper, an intensified SCSO with multiple strategies (IMSCSO) was proposed. The dynamic random search technique was originally used in IMSCSO to improve the convergence efficiency of the algorithm. A hybrid opposition-based learning technique was also created to increase population variety and avoid the algorithm's early convergence. Finally, the joint opposite selection method was developed to strike a balance between the algorithm's exploration and exploitation.

TABLE 16. Details of 23 classical benchmark functions.

No.	Function	D	Domain	Global minimum
UF1	$UF_1(x)=\sum_{i=1}^{D} x_i^2$	30,100,500	[-100,100]	0
UF2	$UF_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30,100,500	[-10,10]	0
UF3	$UF_3(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	30,100,500	[-100,100]	0
UF4	$UF_4(x)=\max_i\left\{\lvert x_i\rvert, 1\le i\le D\right\}$	30,100,500	[-100,100]	0
UF5	$UF_5(x)=\sum_{i=1}^{D-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	30,100,500	[-30,30]	0
UF6	$UF_6(x)=\sum_{i=1}^{D}(\lvert x_i+0.5\rvert)^2$	30,100,500	[-100,100]	0
UF7	$UF_7(x)=\sum_{i=1}^{D} ix_i^4+random[0,1)$	30,100,500	[-1.28,1.28]	0
MF8	$MF_8(x)=-\sum_{i=1}^{D}(x_i\sin(\sqrt{\lvert x_i\rvert}))$	30,100,500	[-500,500]	-418.9829×D
MF9	$MF_9(x)=10D+\sum_{i=1}^{D}\left[x_i^D-10\cos(2\pi x_i)\right]$	30,100,500	[-5.12,5.12]	0
MF10	$MF_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)+20+e$	30,100,500	[-32,32]	0
MF11	$MF_{11}(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30,100,500	[-600,600]	0
MF12	$MF_{12}(x)=\frac{\pi}{D}\{10\sin\pi y_1+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2\pi y_{i+1}]+(y_D-1)^2\}+\sum_{i=1}^{D}U(x_i,10,100,4),\ y_i=1+\frac{x_i+1}{4},$ $U(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	30,100,500	[-50,50]	0
MF13	$MF_{13}(x)=0.1\{\sin^2 3\pi x_1+\sum_{i=1}^{D-1}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_D-1)^2[1+\sin^2(2\pi x_D)]\}+\sum_{i=1}^{D}U(x_i,5,100,4)$	30,100,500	[-50,50]	0
MF14	$MF_{14}(x)=\left[\frac{1}{500}+\sum_{i=1}^{25}\frac{1}{i+\sum_{j=1}^{2}(x_j-a_{j,i})^6}\right]^{-1}$	2	[-65,65]	1
MF15	$MF_{15}(x)=\sum_{i=1}^{D}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
MF16	$MF_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
MF17	$MF_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	0.398
MF18	$MF_{18}(x)=\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]\times\left[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	[-2,2]	3
MF19	$MF_{19}(x)=-\sum_{i=1}^{4}a_i\exp\left(-\sum_{j=1}^{3}b_{ij}(x_j-p_{ij})^2\right)$	3	[1,3]	-3.86
MF20	$MF_{20}(x)=-\sum_{i=1}^{4}a_i\exp\left(-\sum_{j=1}^{6}b_{ij}(x_j-p_{ij})^2\right)$	6	[0,1]	-3.32
MF21	$MF_{21}(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	[0,10]	-10.1532
MF22	$MF_{22}(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	[0,10]	-10.4028
MF23	$MF_{23}(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	[0,10]	-10.5363

TABLE 17. **Details of the CEC-2017 test functions.**

Type	Functions	Global minimum	Domain
Unimodal function	UF17-1 (Shifted and Rotated Bent Cigar Function)	100	[-100,100]
Multimodal functions	MF17-3 (Shifted and Rotated Zakharov Function)	300	
	MF17-4 (Shifted and Rotated Rosenbrock's Function)	400	
	MF17-5 (Shifted and Rotated Rastrigin's Function)	500	
	MF17-6 (Shifted and Rotated Expanded Schaffer's Function)	600	
	MF17-7 (Shifted and Rotated Lunacek Bi_Rastrigin Function)	700	
	MF17-8 (Shifted and Rotated Non-Continuous Rastrigin's Function)	800	
	MF17-9 (Shifted and Rotated Levy Function)	900	
	MF17-10 (Shifted and Rotated Schwefel's Function)	1000	
Hybrid functions	HF17-11 (Zakharov, Rosenbrock, and Rastrigin's Functions)	1100	
	HF17-12 (High Conditioned Elliptic, Modified Schwefel and Bent Cigar Functions)	1200	
	HF17-13 (Bent Ciagr, Rosenbrock and Lunache Bi-Rastrigin Functions)	1300	
	HF17-14 (Elliptic, Ackley, Schaffer and Rastrigin Functions)	1400	
	HF17-15 (Bent Cigar, HGBat, Rastrigin and Rosenbrock Functions)	1500	
	HF17-16 (Expanded Schaffer, HGBat, Rosenbrock and Modified Schwefel Functions)	1600	
	HF17-17 (Katsuura, Ackley, Expanded Griewank plus Rosenbrock, Modifed Schwefel and Rastrigin Functions)	1700	
	HF17-18 (high conditioned Elliptic, Ackley, Rastrigin, HGBat and Discus Functions)	1800	
	HF17-19 (Bent Cigar, Rastrigin, Expanded Grienwank plus Rosenbrock, Weierstrass and Expanded Schaffer Functions)	1900	
	HF17-20 (Happycat, Katsuura, Ackley, Rastrigin, Modified Schwefel and Schaffer Functions)	2000	
Composition functions	CF17-21 (Rosenbrock, High Conditioned Elliptic and Rastrigin Functions)	2100	
	CF17-22 (Rastrigin's, Griewank's and Modified Schwefel's Functions)	2200	
	CF17-23 (Rosenbrock, Ackley, Modified Schwefel and Rastrigin Functions)	2300	
	CF17-24 (Ackley, High Conditioned Elliptic, Girewank and Rastrigin Functions)	2400	
	CF17-25 (Rastrigin, Happycat, Ackley, Discus and Rosenbrock Functions)	2500	
	CF17-26 (Expanded Scaffer, Modified Schwefel, Griewank, Rosenbrock and Rastrigin Functions)	2600	
	CF17-27 (HGBat, Rastrigin, Modified Schwefel, Bent-Cigar, High Conditioned Elliptic and Expanded Schaffer Functions)	2700	
	CF17-28 (Ackley, Griewank, Discus, Rosenbrock, Happy Cat, Expanded Schaffer Functions)	2800	
	CF17-29 (shifted and rotated Rastrigin, Expanded Schaffer and Lunacek Bi_Rastrigin Functions)	2900	
	CF17-30 (shifted and rotated Rastrigin, Non-Continuous Rastrigin and Levy Functions)	3000	

The performance of IMSCSO was validated on 23 classical benchmark functions, 29 CEC 2017 benchmark functions, 10 CEC 2019 benchmark functions, and 10 CEC 2020 benchmark functions. The results obtained by IMSCSO on these benchmark functions were extensively compared with those of 13 well-established optimizers to show IMSCSO's effectiveness. IMSCSO was either better than or roughly similar to its rival optimizers. The statistical findings of the Wilcoxon signed-rank and Friedman tests show that IMSCSO can obtain superior solutions in comparison to 13 rival optimizers. The collected numerical results of IMSCSO on seven constrained engineering design issues also reveal that IMSCSO can provide outstanding results compared to several published optimizers.

In future work, various mutation or acceleration techniques can be used to improve the efficacy of IMSCSO. The binary and multi-objective versions of IMSCSO can also be developed for solving complex problems.

APPENDIX A

See Tables 16–19.

TABLE 18. Details of the CEC 2019 test functions.

No.	Functions	D	Domain	Global minimum
C19-1	Storn's Chebyshev polynomial fitting problem	9	[-8192,8192]	1
C19-2	Inverse Hilbert matrix problem	16	[-16384,16384]	1
C19-3	Lennard-Jones minimum energy cluster	18	[-4,4]	1
C19-4	Rastrigin's Function	10	[-100,100]	1
C19-5	Girewank's Function	10	[-100,100]	1
C19-6	Weierstrass's Function	10	[-100,100]	1
C19-7	Modified Schwefel 's Function	10	[-100,100]	1
C19-8	Expanded Schaffer's F6 Function	10	[-100,100]	1
C19-9	Happy Cat Function	10	[-100,100]	1
C19-10	Ackley Function	10	[-100,100]	1

TABLE 19. Details of the CEC-2020 test functions.

Type	Functions	Global minimum	Domain
Unimodal function	UF20-1 (Shifted and Rotated Bent Cigar Function)	100	[-100,100]
Multimodal functions	MF20-2 (Shifted and Rotated Lunacek Bi_Rastrigin Function)	700	
Hybrid functions	HF20-3 (Function 1 hybrid)	1100	
	HF20-4 (Function 2 hybrid)	1700	
	HF20-5 (Function 3 hybrid)	1900	
	HF20-6 (Function 4 hybrid)	2100	
	HF20-7 (Function 4 hybrid)	1600	
Composition functions	CF20-8 (Function 1 composition)	2200	
	CF20-9 (Function 2 composition)	2400	
	CF20-10 (Function 3 composition)	2500	

REFERENCES

[1] Q. H. Zhai, T. Ye, M. X. Huang, S. L. Feng, and H. Li, "Whale optimization algorithm for multiconstraint second-order stochastic dominance portfolio optimization," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–19, Aug. 2020, doi: 10.1155/2020/8834162.

[2] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1430–1442, Mar. 2021, doi: 10.1109/tcyb.2019.2939219.

[3] B. Gao, H. Yang, H.-C. Lin, Z. Wang, W. Zhang, and H. Li, "A hybrid improved whale optimization algorithm with support vector machine for short-term photovoltaic power prediction," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2014187, doi: 10.1080/08839514.2021.2014187.

[4] M. Tenemaza, S. Lujan-Mora, A. De Antonio, and J. Ramirez, "Improving itinerary recommendations for tourists through metaheuristic algorithms: An optimization proposal," *IEEE Access*, vol. 8, pp. 79003–79023, 2020, doi: 10.1109/ACCESS.2020.2990348.

[5] M. Cavazzuti, "Deterministic optimization," in *Optimization Methods*. Berlin, Germany: Springer, 2013, doi: 10.1007/978-3-642-31187-1_4.

[6] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 115–118, Feb. 2004, doi: 10.1109/LSP.2003.821649.

[7] P. E. Gill and W. Murray, "Quasi-Newton methods for unconstrained optimization," *IMA J. Appl. Math.*, vol. 9, no. 1, pp. 91–108, 1972, doi: 10.1093/imamat/9.1.91.

[8] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis* (Lecture Notes in Mathematics), vol. 630, G. A. Watson, Eds. Berlin, Germany: Springer, 1978, pp. 105–116, doi: 10.1007/BFb0067700.

[9] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[10] H. Stegherr, M. Heider, and J. Hähner, "Classifying metaheuristics: Towards a unified multi-level classification system," *Natural Comput.*, vol. 21, pp. 155–171, 2022, doi: 10.1007/s11047-020-09824-0.

[11] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992. [Online]. Available: https://www.jstor.org/stable/24939139

[12] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.

[13] Bilal, M. Pant, and G. Vig, "Clustering based adaptive differential evolution for numerical optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Glasgow, U.K., Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185874.

[14] Bilal, M. Pant, and V. Snasel, "Design optimization of water distribution networks through a novel differential evolution," *IEEE Access*, vol. 9, pp. 16133–16151, 2021, doi: 10.1109/ACCESS.2021.3052032.

[15] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, Oct. 2006, doi: 10.1109/TEVC.2005.860762.

[16] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, Aug. 1990, doi: 10.1287/inte.20.4.74.

[17] V. Hayyolalam and A. A. Pourhaji Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103249, doi: 10.1016/j.engappai.2019.103249.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, vol. 4, 1995', pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[19] M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, and L. Li, "Phasor particle swarm optimization: A simple and efficient variant of PSO," *Soft Comput.*, vol. 23, pp. 9701–9718, 2019, doi: 10.1007/s00500-018-3536-8.

[20] X. Zhou, H. Ma, J. Gu, H. Chen, and W. Deng, "Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105139, doi: 10.1016/j.engappai.2022.105139.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[23] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[24] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[25] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.

[26] A. Mohammadi-Balani, M. D. Nayeri, A. Azar, and M. Taghizadeh-Yazdi, "Golden eagle optimizer: A nature-inspired metaheuristic algorithm," *Comput. Ind. Eng.*, vol. 152, Feb. 2021, Art. no. 107050, doi: 10.1016/j.cie.2020.107050.

[27] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.

[28] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2019, doi: 10.1016/j.knosys.2018.11.024.

[29] G. Dhiman and A. Kaur, "STOA: A bio-inspired based optimization algorithm for industrial engineering problems," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 148–174, Jun. 2019, doi: 10.1016/j.engappai.2019.03.021.

[30] A. Seyyedabbasi and F. Kiani, "Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems," *Eng. With Comput.*, vol. 39, pp. 1–25, Apr. 2022, doi: 10.1007/s00366-022-01604-x.

[31] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 8, pp. 8457–8482, Aug. 2021, doi: 10.1007/s12652-020-02580-0.

[32] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *J. Supercomput.*, vol. 79, no. 7, pp. 7305–7336, May 2023, doi: 10.1007/s11227-022-04959-6.

[33] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[34] Q. Qi, G. Xu, X. Fan, and J. Wang, "A new specular reflection optimization algorithm," *Adv. Mech. Eng.*, vol. 7, no. 10, 2015, doi: 10.1177/1687814015610475.

[35] B. Ma, P. Lu, L. Zhang, Q. Qi, Y. Chen, Y. Hu, M. Wang, and G. Wang, "CMSRAS: A novel chaotic multi-specular reflection optimization algorithm considering shared nodes," *IEEE Access*, vol. 9, pp. 43050–43094, 2021, doi: 10.1109/ACCESS.2021.3066180.

[36] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190, doi: 10.1016/j.knosys.2019.105190.

[37] M. Abdel-Basset, D. El-Shahat, M. Jameel, and M. Abouhawwash, "Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems," *Comput. Methods Appl. Mech. Eng.*, vol. 403, Jan. 2023, Art. no. 115652, doi: 10.1016/j.cma.2022.115652.

[38] M. Abdel-Basset, R. Mohamed, S. A. A. Azeem, M. Jameel, and M. Abouhawwash, "Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion," *Knowl.-Based Syst.*, vol. 268, May 2023, Art. no. 110454, doi: 10.1016/j.knosys.2023.110454.

[39] M. Abdel-Basset, R. Mohamed, K. M. Sallam, and R. K. Chakrabortty, "Light spectrum optimizer: A novel physics-inspired metaheuristic optimization algorithm," *Mathematics*, vol. 10, no. 19, p. 3466, Sep. 2022, doi: 10.3390/math10193466.

[40] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, and K. M. Sallam, "Fick's law algorithm: A physical law-based algorithm for numerical optimization," *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110146, doi: 10.1016/j.knosys.2022.110146.

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

[42] M. Ghasemi, I. F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, and L. Li, "A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent flow of water-based optimization (TFWO)," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103666, doi: 10.1016/j.engappai.2020.103666.

[43] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.

[44] M. Kumar, A. J. Kulkarni, and S. C. Satapathy, "Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology," *Future Gener. Comput. Syst.*, vol. 81, pp. 252–272, Apr. 2018, doi: 10.1016/j.future.2017.10.052.

[45] S. H. Samareh Moosavi and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," *Eng. Appl. Artif. Intell.*, vol. 86, pp. 165–181, Nov. 2019, doi: 10.1016/j.engappai.2019.08.025.

[46] E. Trojovská and M. Dehghani, "A new human-based metahurestic optimization method based on mimicking cooking training," *Sci. Rep.*, vol. 12, no. 1, p. 14861, Sep. 2022, doi: 10.1038/s41598-022-19313-2.

[47] M. Dehghani and P. Trojovský, "Teamwork optimization algorithm: A new optimization approach for function minimization/maximization," *Sensors*, vol. 21, no. 13, p. 4567, Jul. 2021, doi: 10.3390/s21134567.

[48] E. Pira, "City councils evolution: A socio-inspired metaheuristic optimization algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 14, pp. 1–50, Mar. 2022, doi: 10.1007/s12652-022-03765-5.

[49] M. Dehghani, E. Trojovská, and T. Zuščák, "A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training," *Sci. Rep.*, vol. 12, no. 1, p. 17387, Oct. 2022, doi: 10.1038/s41598-022-22458-9.

[50] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[51] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020, doi: 10.1016/j.ins.2020.06.037.

[52] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, Apr. 2021, Art. no. 113609, doi: 10.1016/j.cma.2020.113609.

[53] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Syst. Appl.*, vol. 181, Nov. 2021, Art. no. 115079, doi: 10.1016/j.eswa.2021.115079.

[54] I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, and A. H. Gandomi, "INFO: An efficient optimization algorithm based on weighted mean of vectors," *Expert Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116516, doi: 10.1016/j.eswa.2022.116516.

[55] M. Abdel-Basset, D. El-Shahat, M. Jameel, and M. Abouhawwash, "Exponential distribution optimizer (EDO): A novel math-inspired algorithm for global optimization and engineering problems," *Artif. Intell. Rev.*, vol. 56, pp. 1–72, Jan. 2023, doi: 10.1007/s10462-023-10403-9.

[56] S. M. Sivalingam, P. Kumar, and V. Govindaraj, "The hybrid average subtraction and standard deviation based optimizer," *Adv. Eng. Softw.*, vol. 176, Feb. 2023, Art. no. 103387, doi: 10.1016/j.advengsoft.2022.103387.

[57] M. Dehghani, Z. Montazeri, O. P. Malik, A. Ehsanifar, and A. Dehghani, "OSA: Orientation search algorithm," *Int. J. Ind. Electron. Control Optim.*, vol. 2, no. 2, pp. 99–112, 2019, doi: 10.22111/ieco.2018.26308.1072.

[58] M. Dehghani, Z. Montazeri, H. Givi, J. Guerrero, and G. Dhiman, "Darts game optimizer: A new optimization technique based on darts game," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 286–294, Oct. 2020, doi: 10.22266/ijies2020.1031.26.

[59] M. Dehghani, M. Mardaneh, J. Guerrero, O. Malik, and V. Kumar, "Football game based optimization: An application to solve energy commitment problem," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 514–523, Oct. 2020, doi: 10.22266/ijies2020.1031.45.

[60] S. Doumari, H. Givi, M. Dehghani, and O. Malik, "Ring toss game-based optimization algorithm for solving various optimization problems," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 545–554, Jun. 2021, doi: 10.22266/ijies2021.0630.46.

[61] B. Ma, Y. Hu, P. Lu, and Y. Liu, "Running city game optimizer: A game-based metaheuristic optimization algorithm for global optimization," *J. Comput. Des. Eng.*, vol. 10, no. 1, pp. 65–107, Jan. 2023, doi: 10.1093/jcde/qwac131.

[62] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 126–142, Apr. 2005, doi: 10.1109/TEVC.2005.843751.

[63] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 2008, pp. 1128–1134, doi: 10.1109/CEC.2008.4630938.

[64] V. T. Aghaei, A. SeyyedAbbasi, J. Rasheed, and A. M. Abu-Mahfouz, "Sand cat swarm optimization-based feedback controller design for nonlinear systems," *Heliyon*, vol. 9, no. 3, Mar. 2023, Art. no. e13885, doi: 10.1016/j.heliyon.2023.e13885.

[65] M. Irfan, N. Ayub, Q. A. Ahmed, S. Rahman, M. S. Bashir, G. Nowakowski, S. M. Alqhtani, and M. Sieja, "AQSA: aspect-based quality sentiment analysis for multi-labeling with improved ResNet hybrid algorithm," *Electronics*, vol. 12, no. 6, p. 1298, Mar. 2023, doi: 10.3390/electronics12061298.

[66] P. Cheng, F. Mao, and H. Zhao, "Relationship study of meltblowning variables based on machine learning algorithm," *Autom. Mach. Learn.*, vol. 3, no. 2, pp. 17–26, 2022, doi: 10.23977/autml.2022.030204.

[67] D. Wu, H. Rao, C. Wen, H. Jia, Q. Liu, and L. Abualigah, "Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems," *Mathematics*, vol. 10, no. 22, p. 4350, Nov. 2022, doi: 10.3390/math10224350.

[68] Y. Li and G. Wang, "Sand cat swarm optimization based on stochastic variation with elite collaboration," *IEEE Access*, vol. 10, pp. 89989–90003, 2022, doi: 10.1109/ACCESS.2022.3201147.

[69] B. Arasteh, A. Seyyedabbasi, J. Rasheed, and A. M. Abu-Mahfouz, "Program source-code re-modularization using a discretized and modified sand cat swarm optimization algorithm," *Symmetry*, vol. 15, no. 2, p. 401, Feb. 2023, doi: 10.3390/sym15020401.

[70] F. Kiani, F. A. Anka, and F. Erenel, "PSCSO: Enhanced sand cat swarm optimization inspired by the political system to solve complex problems," *Adv. Eng. Softw.*, vol. 178, Apr. 2023, Art. no. 103423, doi: 10.1016/j.advengsoft.2023.103423.

[71] D. Jovanovic, M. Marjanovic, M. Antonijevic, M. Zivkovic, N. Budimirovic, and N. Bacanin, "Feature selection by improved sand cat swarm optimizer for intrusion detection," in *Proc. Int. Conf. Artif. Intell. Everything (AIE)*, Lefkosa, Cyprus, Aug. 2022, pp. 685–690, doi: 10.1109/AIE57029.2022.00134.

[72] A. Seyyedabbasi, "A reinforcement learning-based metaheuristic algorithm for solving global optimization problems," *Adv. Eng. Softw.*, vol. 178, Apr. 2023, Art. no. 103411, doi: 10.1016/j.advengsoft.2023.103411.

[73] C. Hamzaçebi and F. Kutay, "Continuous functions minimization by dynamic random search technique," *Appl. Math. Model.*, vol. 31, no. 10, pp. 2189–2198, Oct. 2007, doi: 10.1016/j.apm.2006.08.015.

[74] C. Hamzaçebi and F. Kutay, "A heuristic approach for finding the global minimum: Adaptive random search technique," *Appl. Math. Comput.*, vol. 173, no. 2, pp. 1323–1333, Feb. 2006, doi: 10.1016/j.amc.2005.05.002.

[75] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Automat. (CIMCA), Int. Conf. Intell. Agents, Web Technol. Internet*, vol. 1, 2005, pp. 695–701, doi: 10.1109/CIMCA.2005.1631345.

[76] Z. Wang, H. Ding, Z. Yang, B. Li, Z. Guan, and L. Bao, "Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization," *Appl. Intell.*, vol. 52, pp. 7922–7964, 2022, doi: 10.1007/s10489-021-02776-7.

[77] M. Basu, "Quasi-oppositional differential evolution for optimal reactive power dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 29–40, Jun. 2016, doi: 10.1016/j.ijepes.2015.11.067.

[78] A. Chaudhuri and T. P. Sahu, "Multi-objective feature selection based on quasi-oppositional based Jaya algorithm for microarray data," *Knowl.-Based Syst.*, vol. 236, Jan. 2022, Art. no. 107804, doi: 10.1016/j.knosys.2021.107804.

[79] S. Zhao, Y. Wu, S. Tan, J. Wu, Z. Cui, and Y.-G. Wang, "QQLMPA: A quasi-opposition learning and Q-learning based marine predators algorithm," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119246, doi: 10.1016/j.eswa.2022.119246.

[80] F. Y. Arini, S. Chiewchanwattana, C. Soomlek, and 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.*, vol. 188, Feb. 2022, Art. no. 116001, doi: 10.1016/j.eswa.2021.116001.

[81] F. Y. Arini, K. Sunat, and C. Soomlek, "Golden jackal optimization with joint opposite selection: An enhanced nature-inspired optimization algorithm for solving optimization problems," *IEEE Access*, vol. 10, pp. 128800–128823, 2022, doi: 10.1109/ACCESS.2022.3227510.

[82] Y. Xu, Z. Yang, X. Li, H. Kang, and X. Yang, "Dynamic opposite learning enhanced teaching–learning-based optimization," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 104966, doi: 10.1016/j.knosys.2019.104966.

[83] Q. Fan, Z. Chen, and Z. Xia, "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems," *Soft Comput.*, vol. 24, pp. 14825–14843, Mar. 2020, doi: 10.1007/s00500-020-04834-7.

[84] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113389, doi: 10.1016/j.eswa.2020.113389.

[85] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 372–379.

[86] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan "The 100-digit challenge: problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization," Nanyang Technol. Univ., Tech. Rep., 2018.

[87] J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2020 special session on multimodal multiobjective optimization," Zhengzhou Univ., Tech. Rep., Dec. 2019.

[88] H. Bayzidi, S. Talatahari, M. Saraee, and C.-P. Lamarche, "Social network search for solving engineering optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–32, Sep. 2021, doi: 10.1155/2021/8548639.

[89] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. With Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013, doi: 10.1007/s00366-011-0241-y.

[90] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[91] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013, doi: 10.1016/j.asoc.2012.11.026.

[92] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Inf. Sci.*, vol. 178, no. 15, pp. 3043–3074, Aug. 2008, doi: 10.1016/j.ins.2008.02.014.

[93] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007, doi: 10.1016/j.amc.2006.11.033.

[94] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *Int. J. Gen. Syst.*, vol. 37, no. 4, pp. 443–473, Aug. 2008, doi: 10.1080/03081070701303470.
[95] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014, doi: 10.1016/j.compstruc.2014.03.007.
[96] L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: A novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, vol. 157, Jul. 2021, Art. no. 107250, doi: 10.1016/j.cie.2021.107250.
[97] P. Kim and J. Lee, "An integrated method of particle swarm optimization and differential evolution," *J. Mech. Sci. Technol.*, vol. 23, no. 2, pp. 426–434, Feb. 2009, doi: 10.1007/s12206-008-0917-4.
[98] G. G. Wang, "Adaptive response surface method using inherited Latin hypercube design points," *J. Mech. Design*, vol. 125, no. 2, pp. 210–220, Jun. 2003, doi: 10.1115/1.1561044.
[99] S. Zhao, T. Zhang, S. Ma, and M. Chen, "Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105075, doi: 10.1016/j.engappai.2022.105075.
[100] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937, doi: 10.2307/2279372.
[101] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945, doi: 10.2307/3001968.
[102] M. Abdel-Basset, R. Mohamed, M. Jameel, and M. Abouhawwash, "Spider wasp optimizer: A novel meta-heuristic optimization algorithm," *Artif. Intell. Rev.*, vol. 56, no. 10, pp. 11675–11738, Oct. 2023, doi: 10.1007/s10462-023-10446-y.

XUEWEI LI received the Ph.D. degree from the Huazhong University of Science and Technology, in 2018. He is currently an Instructor with the School of Intelligent Engineering, Henan Institute of Technology, Xinxiang, Henan, China. His research interests include intelligent detection, intelligent robot control, and offline reinforcement learning.

YONGLAN QI received the master's degree from Henan University, in 2021. She is currently an Assistant Lab Master with the School of Intelligent Engineering, Henan Institute of Technology, Xinxiang, Henan, China. Her research interests include machine vision, image processing, and intelligent detection.

QIAN XING received the master's degree from Henan Normal University, in 2014. She is currently an Instructor with the School of Intelligent Engineering, Henan Institute of Technology, Xinxiang, Henan, China. Her research interests include intelligent control, image processing, and machine vision.

YONGTAO HU received the Ph.D. degree from Yanshan University, in 2017. He is currently an Instructor with the School of Electrical Engineering and Automation, Henan Institute of Technology, Xinxiang, Henan, China. His research interests include fault diagnosis and data analysis.