Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085167

ARTICLE

Enhanced Sand Cat with Selective Opposition (ESCSO) Algorithm for Optimization and Engineering Problems

Aisha Tanveer[1], **Noraini Ibrahim**[1], **Muhammad Zubair Rehman**[2,*], **Abdullah Khan**[3] **and Nazri Mohd. Nawi**[1]

[1]Soft Computing & Data Mining Centre (SMC), Faculty of Computer Science & Information Technology (FSKTM), Universiti Tun Hussein Onn Malaysia, Parit Raja, Malaysia

[2]Faculty of Computing & IT, Sohar University, Sohar, Oman

[3]Institute of Computer Science and Information Technology, The University of Agriculture Peshawar, Peshawar, Pakistan

*Corresponding Author: Muhammad Zubair Rehman. Email: sgillani@su.edu.om

Received: 06 May 2026; Accepted: 11 June 2026; Published: 13 August 2026

ABSTRACT: Metaheuristic optimization algorithms have gained wide adoption in engineering and scientific domains. However, many swarm-based methods struggle to balance exploration and exploitation, often converging prematurely on suboptimal solutions. The Sand Cat Swarm Optimization (SCSO) algorithm is one such method, with limited exploration ability constraining its performance on complex problem landscapes. This paper introduced the Enhanced Sand Cat with Selective Opposition (ESCSO) algorithm which combines opposition-based learning with a velocity mechanism to overcome this limitation. In ESCSO, under-performing candidates referred to as Sigma-variant cats are identified using Spearman correlation and replaced with their opposite solutions to inject diversity into the search process. Stronger candidates termed Sigma cats, act as elite guides pulling the search toward better regions. A PSO-inspired velocity update governs both roles, keeping exploration and exploitation in balance rather than letting one dominate. Tested across 30 benchmark functions plus two real engineering problems, reflectarray antenna design and microgrid energy management, ESCSO achieves competitive convergence, solution quality, and robustness when compared to recent state-of-the-art methods.

KEYWORDS: Opposition-based learning; metaheuristics; search algorithm; sand cat optimization; swarm intelligence; microgrid energy management; Reflectarray antenna

1 Introduction

Rapid advancements in computing have addressed numerous engineering problems but have also introduced complex challenges, questioned traditional methodologies and necessitated innovative solutions [1]. These challenges frequently emerge as complex problems within unpredictable solution spaces, necessitating the exploration of unconventional approaches. In recent years, metaheuristic algorithms have emerged as adaptive tools for addressing these challenges [2]. These algorithms effectively balance flexibility, simplicity, and robustness while delivering approximate solutions to computationally complex problems within a reasonable time [3]. Their iterative nature emphasizes efficiency and effectiveness over exact optimality. In recent years, several types of metaheuristic algorithms inspired by nature, mimicking human behavior, or physical events have been introduced. These categories include swarm-based, evolutionary-based, physics-based, and human-based algorithms. Swarm-based algorithms, such as Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO), emulate the behavior of biological groups [4]. The Genetic Algorithm

(GA) and Differential Evolution (DE) are two examples of evolutionary-based algorithms that mimic the way evolution works in nature [5]. Simulated Annealing (SA) and Gravitational Search Algorithm (GSA) are physical-based algorithms that work by copying the rules of physics or natural phenomena [6]. Human-based algorithms, like Teaching Learning-Based Optimization (TLBO) and Harmony Search (HS), derive inspiration from human behavior and societal structures [7]. Recently, a new addition to the realm of swarm-based metaheuristic algorithms is the Sand Cat Swarm Optimization (SCSO) algorithm. Inspired by the hunting behavior of cats, SCSO leverages the social interactions and coordination observed in feline communities [8]. This algorithm offers a distinctive approach to problem-solving, effectively leveraging the dynamic nature of cat swarms to explore solution spaces. The SCSO algorithm's effectiveness lies in its ability to emulate the collaboration and communication behavior observed among sand cats during hunting activities. The SCSO algorithm has successfully demonstrated its effectiveness in solving several NP-hard engineering problems. Although improvements to SCSO algorithm have enhanced its performance, issues such as slow convergence and entrapment in local optima persist due to limited exploration ability [9]. Therefore, this paper proposes an enhanced sand cat with a selective opposition algorithm (ESCSO) that will avoid local optima during exploration due to its opposition learning mechanism. The main contributions of this paper are as follows:

(1) The paper proposes a novel Enhanced sand cat with selective opposition (ESCSO). ESCSO works by integrating selective Opposition-Based Learning with velocity updates from PSO.
(2) To further improve ESCSO, a structured mechanism is introduced using Spearman's correlation to apply opposition learning only to underperforming agents (Sigma-variant cats), guided by elite solutions (Sigma cats). This approach maximizes search efficiency, reduces unnecessary evaluations, and improves solution precision by guaranteeing that opposition is only applied when needed. The introduction of selective opposition, fitness-driven leadership, and adaptive velocity updates makes the ESCSO algorithm highly effective for complex optimization problems.
(3) ESCSO is extensively validated on 30 benchmark functions based on the Mirjalili [10] framework and two real-world applications, i.e., Reflectarray antenna optimization and microgrid energy management.
(4) Statistical evaluation using the Friedman test show that ESCSO performs significantly better than existing state-of-the-art metaheuristics on a majority of benchmark functions in terms of global convergence.

The rest of the paper is organized as follows: Section 2 sheds some light on the literature review; then the original SCSO algorithm and the proposed ESCSO algorithm is discussed in Section 3. Section 4 presents the results and discussions, Section 5 applies ESCSO to two engineering problems; and Section 6 concludes the paper and outlines future work.

2 Related Work

Recent years have witnessed a surge in research and application of metaheuristic algorithms, propelled by advances in computational resources. This has enabled the exploration of more complex algorithms, the hybridization of existing ones, and the development of variant algorithms. Nature inspired metaheuristic algorithms simulate natural phenomena and build mathematical models, contributing to problem-solving in various domains. Metaheuristic algorithms can be broadly categorized into Biology-based, Nature-based, and Physics-based algorithms. These algorithms aim to find solutions close to the optimal solution in a global search domain with minimal time and cost overhead. Among biology-based algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization algorithm (PSO) are widely used. GA simulates biological evolution, while PSO originates from the predatory behavior of bird flocks [11]. Numerous other biology-based algorithms, such as Differential Evolution (DE) [12], Grey Wolf Optimization (GWO) [13], Artificial Bee Colony (ABC) [14], Manta Ray Foraging [15], and African Vultures Optimization [16], draw inspiration from various aspects of the natural world. Aptenodytes Forsteri Optimization Algorithm (AFO) [17], Human Behavior-Based Optimization Algorithm (HBBOA) [18], and Turbulent Flow of Water-based Optimization (TFWO) [19] are examples of nature-based algorithms that simulate natural laws governing plants, water, and more. Physics-based meta-heuristic algorithms, such as Galaxy-based Search Algorithm (GbSA) [20], Electric Charged Particles Optimization (ECPO) [21], and Henry Gas Solubility Optimization (HGSO) [22], simulate physical theories and phenomena.

The Sand Cat Swarm Optimization (SCSO) algorithm, introduced by Amir Seyyedabbasi in 2022, is a biology-based algorithm inspired by the predatory behavior of sand cats [8]. Known for its simple structure, easy implementation, and minimal parameters, SCSO simulates the hunting behavior of sand cats in desert environments. It incorporates auditory characteristics, gradually decreasing sensitivity as the sand cat approaches its prey, enhancing exploitation efficiency [8]. Despite its merits, SCSO exhibits certain limitations, prompting researchers to propose improvements. In 2022, it was reported that the sand cat has a very strong exploitation ability when approaching prey, but it is prone to fall into the local optimum due to poor exploration ability. Therefore, a modified sand cat swarm optimization (MSCSO) algorithm with a walking strategy and a lens opposition-based learning with Triangle Walking (TW) strategy was proposed to offer a stronger exploration ability. The Roulette Wheel Selection was used to obtain the walking step length. MSCSO algorithm increased the exploration ability of the SCSO algorithm and made the MSCSO algorithm more global convergent [9]. Some metaheuristics have incorporated low level hybridization. In selective opposition research, Dhargupta et al. [23] demonstrated that opposition applied to carefully chosen wolf agents yields faster, more reliable convergence. In 2025, Praghya Mehrotra and Diwakar Bhardwaj proposed a Hybrid Grey Wolf Optimizer (HGWO) for energy-aware clustering and routing in wireless sensor networks, showing that targeted diversification mechanisms can substantially improve coverage and energy balance [24]. More recently, Subburathinam et al. [25] introduced Hybrid Grey Wolf Optimizer-Harmony Search Algorithm (Hybrid GWO-HSA) algorithms for smart wireless sensor networks, emphasizing the role of adaptive exploration to pinpoint factors affecting node communication. Inspired by these findings, ESCSO offers a different approach that works by (i) identifying underperforming agents via Spearman correlation, (ii) applying selective opposition only to the underperforming agents, (iii) guiding exploitation through elite Sigma cats, and (iv) stabilizing trajectories with PSO-inspired velocity updates.

Li and Wang [26] introduced an elite partnership and stochastic search mechanism to improve the convergence accuracy of SCSO. Iraji et al. [27] diversified the initial population of SCSO with chaotic sequences and integrated it with a pattern search to assess the minimum safety factor of slopes. Jovanovic et al. [28] integrated the Artificial Bee Colony (ABC) algorithm with SCSO to enhance its local optima avoidance ability and to optimize intrusion detection in networks. Wu et al. [29] added a wandering strategy to improve SCSO's ability to jump out of local optima. Seyyedabbasi [30] successfully applied SCSO to solve the position and rotation of an industrial robotic arm. Kiani et al. [31] used random selection of a new position between the optimal candidate solution and the current position which resulted in enhanced search in SCSO algorithm. Seyyedabbasi [30] integrated reinforcement learning with SCSO to further enhance the algorithm's convergence on global optimization problems. The summary of some of the important contributions made to the SCSO is given in Table 1.

Table 1: Important contributions to SCSO algorithm.

Number	Reference	Algorithm(s)	Application(s)
[8]	Seyyedabbasi and Kiani (2022)	SCSO	Global Optimization Problems
[26]	Li and Wang (2022)	SCSO with Stochastic Variation and Elite Collaboration	Improved Convergence Accuracy
[27]	Iraji et al. (2022)	Hybrid Chaotic SCSO with Pattern Search	Minimum Safety Factor Evaluation of Slopes
[28]	Jovanovic et al. (2022)	SCSO with ABC	Hybrid Intrusion Detection System Optimization
[29]	Wu et al. (2022)	Modified SCSO	Constrained Engineering Optimization Problems
[30]	Seyyedabbasi (2022)	SCSO	Inverse Kinematics of Robot Arms
[31]	Kiani et al. (2023)	PSCSO	Complex Problem Solving using Political System
[32]	Seyyedabbasi (2023)		SCSO with Reinforcement Learning Global Optimization Problems

Although the improvements to the SCSO algorithm have enhanced its performance, there are still some problems like slow convergence and getting stuck in local optima due to its poor exploration ability [9]. Therefore, this research proposes ESCSO algorithm that will integrate opposition-based learning and a structured velocity update inspired by Particle Swarm Optimization (PSO) to overcome the original Sand Cat Swarm Optimization (SCSO) algorithm's limited exploration and poor convergence. ESCSO introduces selective opposition for under-performing solutions, identified as sigma-variant cats through Spearman's correlation coefficient and guides the search via elite agents called the Sigma cats.

3 Sand Cat Swarm Optimization (SCSO) Algorithm

Cats are resilient animals capable of adapting easily to various environments. A sand cat also follows the similar traits of the normal cats, but the harsh environment of the desert has enhanced its sense of survival and prey. Its hearing and hunting abilities at night have become more efficient than a normal cat; and it can cover more distance than the normal cats moving ever more precisely towards their prey by utilizing their sense of low frequency sounds detection. Although the Sand cat is a lone hunter in the desert but in the Sand Cat Swarm Optimization (SCSO) algorithm [8], a swarm of cats searching (exploration) and attacking (exploitation) prey is considered. In SCSO algorithm, each sand cat SC_n is considered a search agent moving in any arbitrary direction to solve a fitness function, as given in the Eq. (1).

$$f(SC_n) = f(SC_1, SC_2, SC_3, \ldots, SC_N) \tag{1}$$

$$\vec{r}_G = S - \frac{S \cdot t}{T} \tag{2}$$

Here, $\vec{r}_G$ is a sensitivity parameter of Sand cat that guides R in switching the SCSO from exploration to exploitation phase. It starts from 2 and linearly decreases to 0 as the iteration progresses. S is a constant with a value of 2. T indicates maximum number of iterations and t indicates current iteration.

$$\vec{R} = 2 \cdot \vec{r}_G \cdot \text{rand}(0,1) - \vec{r}_G \tag{3}$$

The $\vec{R}$ obtained from the Eq. (3) enables SCSO to find a fine balance between exploration and exploitation phases adaptively. In each iteration sensitivity range of each cat is calculated using the Eq. (4).

$$\vec{r} = \vec{r}_G \cdot \text{rand}(0,1) \tag{4}$$

The behavior of SCSO searching and switching positions between exploration and exploitation behavior in each iteration is represented in Eq. (5).

$$\vec{X}(t+1) = \begin{cases} \vec{X}_b(t) - \vec{X}_r \cdot \cos(\theta) \cdot \vec{r}, & \text{if } |\vec{R}| \le 1 \quad \text{(exploitation)} \\ \vec{r} \cdot \left(\vec{X}_c(t) - \vec{X}_k(t) \cdot \text{rand}(0,1)\right), & \text{if } |\vec{R}| > 1 \quad \text{(exploration)} \end{cases} \tag{5}$$

Here, $\vec{X}_k$ represents the current position of the cats, $\vec{X}_c$ represents the best candidate cat location, $\vec{X}_r$ points to a random position, and $\vec{X}_b$ represents the position of the global best cat. The angle θ used within a cosine function plays a role in bringing the cat closer to the prey. Initially, the cats keeps on looking for new possible solutions in the global area, but when the condition $|\vec{R}| \le 1$ becomes true, the algorithm sends the cats towards exploitation or attacking the prey. The θ value is selected based on Roulette Wheel [8].

3.1 The Enhanced Sand Cat with Selective Opposition (ESCSO) Algorithm

Opposition-Based Learning (OBL) introduces a mathematical framework inspired by the concept of opposition in nature to enhance the performance of optimization algorithms. The primary goal is to balance exploration and exploitation by considering both the estimate solution X and its opposite X' in the search space [31]. This is formalized as:

$$X' = X_{\min} + X_{\max} - X \tag{6}$$

where, $X_{\min}$ and $X_{\max}$ are the lower and upper bounds of the search space. After generating the opposite solution X' via Eq. (6), both the candidate solution X and its opposite X' are evaluated, and the fittest of the two is retained. For a minimization problem, the update rule is:

$$X \leftarrow \begin{cases} X', & \text{if } f(X') < f(X) \\ X, & \text{otherwise} \end{cases} \tag{7}$$

Inspired by the selective opposition mechanism of Grey Wolf optimization [31], in the proposed ESCSO, strongest cats are termed as sigma cats (σ) and the under-performing, identified by the rank correlation below, are designated as sigma-variant cats (ϵ). The sigma cats guide exploration, while sigma-variant cats are replaced by their opposite solutions to enhance diversity. Then, PSO's velocity update is integrated into SCSO algorithm for updating the positions of the cats in the search space. The Sigma (σ) and Sigma-variant (ϵ) cats are selected based on the highest fitness values during exploration.

$$X_\sigma = \arg \min_{i \in \{1,\ldots,N\}} f(X_i) \tag{8}$$

where X_σ represents the position of the Sigma sand cat.

$$S_\epsilon = \{X_i \mid \rho_i < \rho_{\text{th}},\ i \neq \sigma\} \tag{9}$$

where S_ϵ is the set of Sigma-variant cats, ρ_i is the Spearman rank correlation between agent X_i and the Sigma cat X_σ (Eq. (12)), and ρ_{th} is the opposition threshold. To enhance convergence, velocity update parameter of the PSO algorithm is integrated into the ESCSO.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 \left(X_{\text{best},i} - X_i(t)\right) + c_2 r_2 \left(X_{\text{global}} - X_i(t)\right) \tag{10}$$

where $V_i(t)$ is the velocity of i-th sand cat at iteration t, ω is the inertia weight, c_1 and c_2 are acceleration coefficients, $r_1, r_2 \sim U(0,1)$ are random variables, $X_{\text{best},i}$ is the personal best position of the i-th cat, and X_{global} is the global best position. The new position of the sand cat is updated to show the movements of sand cats towards more promising solutions.

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{11}$$

In the proposed ESCSO algorithm, the Spearman correlation coefficient (ρ) is used to measure the association between the sigma sand cat and the sigma-variant sand cats. Considering $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$, the correlation is computed as:

$$\rho = 1 - \frac{6\sum_{j=1}^{n} d_j^2}{D(D^2-1)} \tag{12}$$

The rank difference between the corresponding sigma and sigma-variant sand cats is defined as:

$$d_j = \text{rank}(x_{i,j}) - \text{rank}(x_{\sigma,j}) \tag{13}$$

where d_j is the difference between the rank of the j-th coordinate of agent X_i and the rank of the j-th coordinate of the Sigma cat X_σ, with ranks taken over the D dimensions.

A high ρ_i (close to 1) means the agent's coordinates are ordered similarly to the elite Sigma cat, i.e., the agent already follows the leading search direction. A low ρ_i indicates a weakly aligned (under-performing) agent. Agents whose ρ_i falls below the threshold ρ_{th} are therefore designated Sigma-variant cats and diversified through selective opposition. This selective rule, inspired by the selective-opposition Grey Wolf Optimizer [23], confines opposition to the agents that most need diversification and avoids disturbing agents that are already converging well. This mechanism enhances exploration by introducing opposite solutions in dimensions exhibiting weak correlation, thus improving the balance between exploration and exploitation in the ESCSO algorithm. In ESCSO, initialization process involves setting up the population based on the specified parameters, including the number of search agents, upper and lower bounds, and the dimension of the search agents. Subsequently, the population vectors are randomly selected while adhering to the predefined constraints. The proposed ESCSO algorithm is illustrated in Fig. 1, while the complete workflow integrating the SCSO search behavior, the PSO-based velocity update, and the selective opposition-based strategy is presented in Algorithm 1.

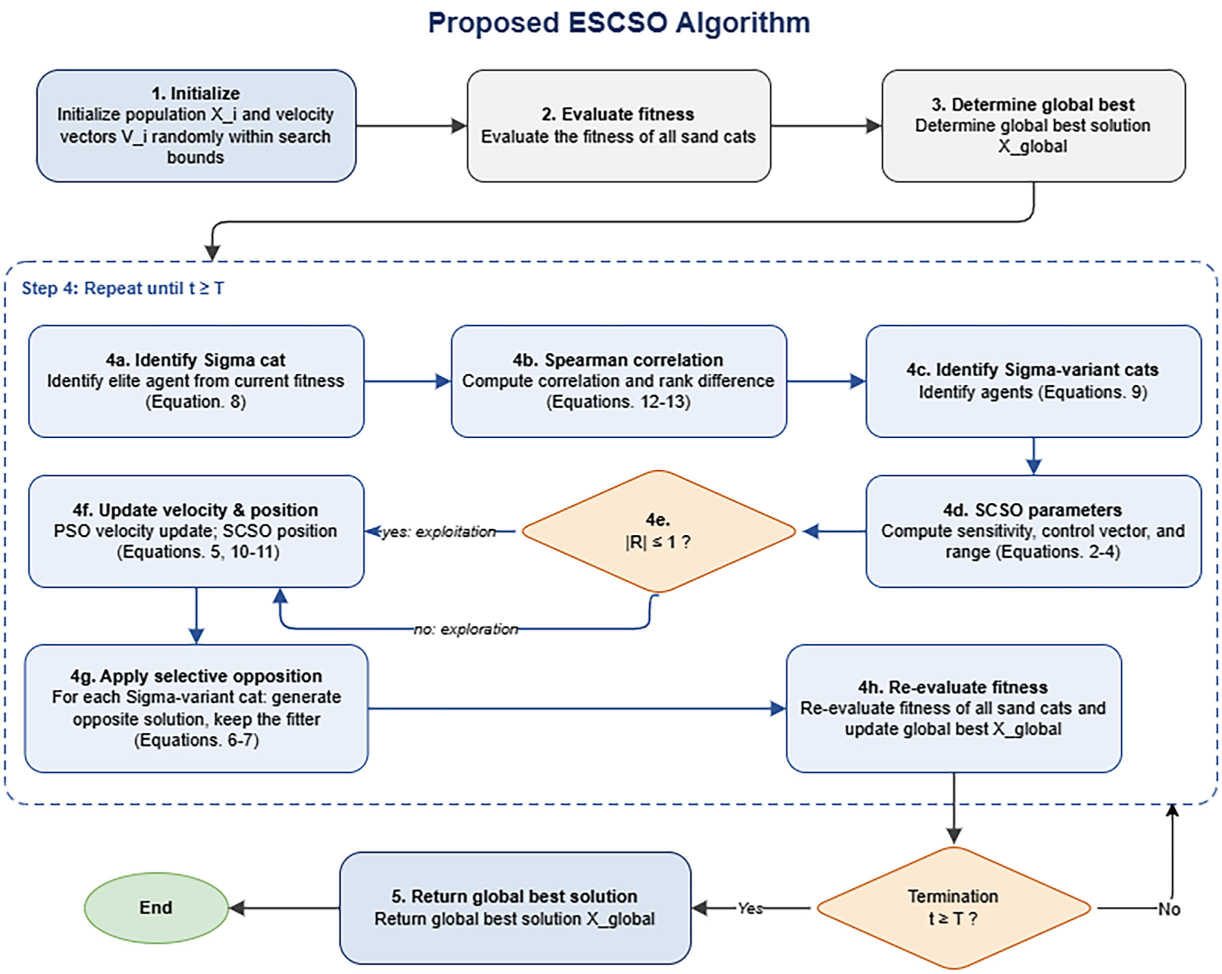

Figure 1: Flowchart of the proposed enhanced sand cat with selective opposition (ESCSO) algorithm.

Algorithm 1: Enhanced sand cat swarm optimization (ESCSO)

Require: Population size N, dimension D, maximum iterations T
Ensure: Global best solution X_{global}
1: Initialize sand cat population X_i randomly within search bounds
2: Initialize velocity vectors V_i
3: Evaluate fitness of all sand cats using Eq. (1)
4: Determine global best solution X_{global}
5: Identify Sigma cats X_σ using Eq. (8)
6: Identify Sigma-variant cats X_ϵ using Eq. (9)
7: **for** $t = 1$ to T **do**
8: Compute sensitivity parameter using Eq. (2)
9: Compute control vector using Eq. (3)
10: **for** each sand cat $i = 1$ to N **do**
11: Compute sensitivity range using Eq. (4)
12: **if** $|\vec{R}| \leq 1$ **then**

(Continued)

Algorithm 1 (continued)

Perform exploitation phase using Eq. (5)
else
Perform exploration phase using Eq. (5)
end if
Update velocity using Eq. (10)
Update position using Eq. (11)
end for
Compute Spearman correlation coefficient using Eq. (12)
Compute dimensional opposition using Eq. (13)
Identify underperforming Sigma-variant cats
for each Sigma-variant cat **do**
Generate opposite solution using Eq. (6)
Apply selective opposition using Eq. (7)
end for
Re-evaluate fitness of all sand cats
Update Sigma and Sigma-variant cats
Update global best solution X_{global}
end for
return

4 Results and Discussions

In this section, the proposed ESCSO algorithm is tested on 30 benchmark functions and its performance is compared with nine other state-of-the-art algorithms. The detailed implementation, parameter settings, and results are discussed in the sub-sections.

4.1 Implementation of ESCSO and the Benchmarked Algorithms

The performance of the ESCSO algorithm were evaluated on the 30 benchmark functions employed by Seyyedabbasi and Kiani [8]. The algorithms were implemented on a Windows 10-based Python environment and ran on an AMD Ryzen 5 processor with 16 GB of RAM. To assess efficiency, the proposed ESCSO was compared with the latest algorithms listed in Table 2. The evaluation involved three types of benchmark functions: unimodal, multimodal, and hybrid each described in detail in the sub-sections.

Table 2: Initialization parameter values for the comparative algorithms.

Algorithm	Parameters
Ant Lion Optimizer (ALO)	NP = 50; Elite Antlion = Adaptive
Artificial Bee Colony (ABC)	NP = 50; $n_e = \frac{NP}{2}$, $n_o = \frac{NP}{2}$
Firefly Optimization (FFO)	NP = 50; $\beta_0 = 1$; Distance = Euclidean
Sand Cat Swarm Optimization (SCSO)	NP = 50; $P_{esc} = 0.2$; $\alpha \in [2, 0]$; $\beta = 1.5$
Grey Wolf Optimizer (GWO)	NP = 50; $\alpha \in [2, 0]$; X = Random initialization
Harris Hawk Optimization (HHO)	NP = 50; $r_1, r_2 \in [0, 1]$; X = Random initialization
Moth Flame Optimization (MFO)	NP = 50; X = Random initialization; F = Best solution per iteration
Particle Swarm Optimization (PSO)	NP = 50; $\omega \in [0.7, 1.4]$; $c_1 \in [1.5, 2.0]$; $c_2 \in [1.5, 2.0]$
Whale Optimization Algorithm (WOA)	NP = 50; $C \in [0.5, 1.0]$

4.2 Parameter Settings

Throughout the experiments, default parameters were employed for all comparative algorithms. The maximum iterations for all algorithms were set to 1000 iterations to ensure a fair comparison between them. A total of 30 trials were conducted on each function and for each algorithm. The algorithms are terminated when either the maximum iteration is reached or the target fitness is achieved. Table 2 shows all the parameter settings of all algorithms used for comparison in this paper. To evaluate the robustness of ESCSO, a sensitivity analysis was conducted on its key parameters: inertial weight (w), cognitive coefficient (c_1), social coefficient (c_2), and opposition threshold (ρ). Each parameter was varied while keeping the others constant. The results indicated that ESCSO maintains high performance across a reasonable parameter range, with optimal setting found at $w = 0.7$, $c_1 = 1.5$, $c_2 = 2.0$, and $\rho = 0.3$.

For statistical analysis, the Standard Deviation (SD) is employed to identify variations in the average trial values, and the Mean formula is utilized to calculate the average across all trials. The equations for *SD* and *Mean* are expressed as follows [32]:

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{14}$$

$$\bar{x} = \frac{x_1 + \cdots + x_n}{n} \tag{15}$$

where n represents the total number of values or elements.

4.3 Description of the Benchmark Functions

The performance of the ESCSO algorithm is assessed using the 30 mathematical functions employed by Mirjalili et al. [10]. To test the algorithms, benchmark functions are categorized into three types: Unimodal, multimodal, and hybrid functions. The detailed descriptions of the function and their properties are given in Table 3, respectively. Dimension, Search Range, and $f_{\min}$ column in the Table 3 denote the dimensions of the function, boundary of the search space, and the cost of the function.

Table 3: Formulae, search range, and minimum value of benchmark functions.

Function (F)	Formulae	Search Range	$f_{\min}$
	Unimodal Functions		
Sphere (F01)	$f(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$	0
Sum of Squares (F02)	$f(x) = \sum_{i=1}^{n} \left(10^6\right)^{\frac{i-1}{n-1}} x_i^2$	$[-10, 10]$	0
Ellipsoid (F03)	$f(x) = \sum_{i=1}^{n} i x_i^2$	$[-100, 100]$	0
Convex (F04)	$f(x) = \sum_{i=1}^{n} (x_i - i)^2$	$[-100, 100]$	0
Quartic (F05)	$f(x) = \sum_{i=1}^{n} x_i^4$	$[-1, 1]$	0
Bent Cigar (F06)	$f(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$	$[-100, 100]$	0
Discus (F07)	$f(x) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$	$[-100, 100]$	0
Shifted Rosenbrock (F08)	$f(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2\right]$	$[-5, 10]$	0
Shifted Griewank (F09)	$f(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]$	0
	Multimodal Functions		
Ackley (F10)	$f(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum x_i^2}} - e^{\frac{1}{n}\sum \cos(2\pi x_i)} + 20 + e$	$[-32.76, 32.76]$	0
Rastrigin (F11)	$f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	0

(Continued)

Table 3 (continued)

Function (F)	Formulae	Search Range	$\mathbf{f_{min}}$
Rosenbrock (F12)	$f(x) = \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-5, 10]$	0
Schwefel (F13)	$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]$	0
Griewank (F14)	$f(x) = \frac{1}{4000}\sum x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0
Levy (F15)	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i - 1)^2[1 + 10\sin^2(\pi w_{i+1})] + (w_n - 1)^2[1 + \sin^2(2\pi w_n)]$	$[-10, 10]$	0
Michalewicz (F16)	$f(x) = -\sum \sin(x_i)\left[\sin\left(\frac{i x_i^2}{\pi}\right)\right]^{2m}$	$[0, \pi]$	−1.8013
Ackley Shifted (F17)	$f(x) = -ae^{-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - o_i)^2}} + e^{\frac{1}{n}\sum_{i=1}^{n}\cos(c(x_i - o_i))} + a + e$	$[-32.76, 32.76]$	0
Rastrigin Shifted (F18)	$f(x) = 10n + \sum_{i=1}^{n}\left[(x_i - o_i)^2 - 10\cos(2\pi(x_i - o_i))\right]$	$[-5.12, 5.12]$	0
Alpine (F19)	$f(x) = \sum_{i=1}^{n}\lvert x_i \sin(x_i) + 0.1x_i \rvert$	$[-10, 10]$	0
McCormick (F20)	$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$	$x_1 \in [-1.5, 4]$, $x_2 \in [-3, 4]$	−1.9133
Egg Holder (F21)	$f(x) = -(x_2 + 47)\sin\left(\sqrt{\left\lvert x_2 + \frac{x_1}{2} + 47 \right\rvert}\right) - x_1 \sin\left(\sqrt{\lvert x_1 - (x_2 + 47) \rvert}\right)$	$[-512, 512]$	−959.6407
Step (F22)	$f(x) = \sum_{i=1}^{n}\lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]$	0
Weierstrass (F23)	$f(x) = \sum_{i=0}^{\infty} a^i \cos(b^i \pi x)$	$[-0.5, 0.5]$	0
Shifted Ackley (F24)	$f(x) = -20e^{-0.2\sqrt{\frac{1}{D}\sum x_i^2}} - e^{\frac{1}{D}\sum\cos(2\pi x_i)} + 20 + e$	$[-32.76, 32.76]$	0
Compositional (F25)	$f(x) = \sum_{i=1}^{D}\sum_{j=1}^{i}(x_i - j)^2$	$[-5, 5]$	0
Katsuura (F26)	$f(x) = \prod_{i=1}^{n}\left[1 + (i+1)\sum_{k=1}^{d}\lfloor 2^k x_i \rfloor 2^{-k}\right]$	$[-5, 5]$	0
Hybrid Functions			
Expanded Schwefel (F27)	$f(x) = \sum_{i=1}^{n-1} g(x_i, x_{i+1}) + g(x_n, x_1),\ g(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$	$[-500, 500]$	0
Schwefel 2.21 (F28)	$f(x) = \sum_{i=1}^{n}\lvert x_i \rvert + \prod_{i=1}^{n}\lvert x_i \rvert$	$[-500, 500]$	0
Schwefel 2.22 (F29)	$f(x) = \sum_{i=1}^{n}\lvert x_i \rvert^{\alpha}$	$[-500, 500]$	0
Shuffled Schwefel (F30)	$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert}) + \sum_{i=1}^{d} x_i^2$	$[-500, 500]$	0

4.4 Simulations Results on Benchmark Functions

Depending on their complexity, the benchmark functions are divided into four types: Unimodal (F01–F09), multimodal (F10–F23), and hybrid (F24–F30), as shown in Table 3. Unimodal functions have no local optima, a single global optimum and are used to evaluate the exploitation capability of the algorithm. Multimodal functions have multiple local optima, a single global optimum and are used to measure the exploration capability of the algorithm. Meanwhile, the hybrid functions are used to find the fine balance between exploration and exploitation capability of an algorithm. The statistical mean values, best values, and standard deviation for Unimodal functions are given in Table 4.

The algorithms' performance on benchmark functions F01 through F09 were evaluated based on their best, average, and standard deviation values. ESCSO performed competitively across the unimodal functions, ranking among the best methods on most of them. It was not uniformly best on every function, for PSO and MFO achieved lower average error on F01, and GWO performed strongly on F06–F07. ESCSO demonstrated a strong ability to identify global minimum, achieving optimal best values of zero for F01 through F04, comparable to algorithms such as GWO and HHO. Conversely, ALO and ABC generated inadequate solutions, while PSO and FFO yielded competitive but inconsistent outcomes. Furthermore, ESCSO demonstrated its ability to maintain stability and consistency in convergence across multiple trials by performing exceedingly well in terms of average values and standard deviation. ESCSO was able to maintain its competitive advantage by consistently achieving low average and standard deviation values and near-optimal best values for F05. Although ALO and FFO suffered from increasing unpredictability and less reliable outcomes, other algorithms including ABC, MFO, and GWO showed promising results. GWO and MFO achieved the strongest performance on F06–F07; although, ESCSO was competitive on F06 but showed a higher average error on F07 in multiple trials. Conversely, ALO, ABC, and FFO demonstrated inconsistent convergence patterns and significantly higher best values. F08 results further emphasized the efficacy of ESCSO which demonstrated exceptional reliability and produced nearly zero best values. ESCSO's answers were more stable, even though ALO and PSO did well. Lastly, for F09, both methods (MFO, and ESCSO) achieved ideal or near-optimal results; nevertheless, ESCSO's low standard deviation and average values made it stand out once again. Yet FFO and ALO were hindered by increased unpredictability and more frequent deviations from the global minimum. Fig. 2 shows the average convergence values for each algorithm on functions, F01–F09 for 30 trial runs. ESCSO achieves top performance across most functions, reaching optimal or near-optimal solutions faster than WOA and MFO in most cases. The average results suggest ESCSO's robustness in maintaining convergence speed and precision.

Table 4: Best, average, and standard deviation values of algorithms on unimodal functions.

F	Metric	ALO	ABC	FFO	PSO	GWO	HHO	MFO	WOA	SCSO	ESCSO
F01	Best	0.017942	5.49E−57	4.66E−05	6.71E−229	0	6.89E−103	0	0	5.54E−08	0
	Avg	0.056367	3.27114	7.210976	0.035869	0.128788	0.07231	0.055842	0.315774	16.82275	0.07215
	SD	0.226484	18.25117	19.21135	0.683393	3.419078	1.561846	0.999697	8.017677	5.67023	1.212323
F02	Best	7.05E−05	1.31E−16	2.25E−04	1.36E−229	0	2.38E−102	0	0	7.06E−07	0
	Avg	0.010342	0.034712	0.004199	0.002486	0.001876	0.000655	0.002553	0.001335	0.968158	0.001121
	SD	0.075123	0.455434	0.10446	0.048333	0.034883	0.016103	0.062837	0.029536	3.395193	0.012139
F03	Best	0.003456	0	2.64E−06	2.22E−241	0	3.63E−218	0	0	2.46E−05	0
	Avg	0.332612	0.034514	25.94896	0.170145	0.002166	0.136365	0.437304	0.000522	32.89717	0.000124
	SD	3.049022	0.086512	139.1934	3.730382	0.054681	2.992451	13.72411	0.01605	101.6486	0.01648
F04	Best	0.007189	2.72E−131	11.85889	1.89E−175	0	4.17E−72	0	0	2.99E−05	0
	Avg	0.193876	0.418616	13.4038	0.016966	0.113675	0.246259	0.163871	0.04542	16.74585	0.022322
	SD	1.266944	2.512458	1.327147	0.237427	2.788138	4.293883	2.704047	0.990408	56.3395	0.963919
F05	Best	0.001252	6.49E−05	0.393326	1.23E−04	0.001875	0.008152	8.45E−05	0.002849	0.000991	0.000398
	Avg	0.002623	0.001177	0.150512	0.000694	0.028894	0.00858	0.030864	0.509647	0.543917	0.067358
	SD	0.003675	0.003956	0.105455	0.002839	0.030253	0.00438	0.032989	0.291641	0.069501	0.004651
F06	Best	39.21053	705.0271	3.159403	2.33E−222	0	2.71E−100	0	0	0	0
	Avg	110.4319	91183.13	889815.8	253.2935	408.1102	723.3784	4.536005	17.65536	14.187511	19.565551
	SD	218.2726	251639	4096911	4551.347	8553.739	12268.24	79.74603	393.0896	121.0908	151.12
F07	Best	31.68146	213.9441	125.7479	2.79E−222	0	6.3E−100	0	0	54.32337	84.32337
	Avg	522.7827	515146.9	553740.2	459.3747	107.1906	54.43745	142.6669	78.32644	30125619	15742999
	SD	5254.64	1990481	2235866	14299.18	2538.465	272.1716	3743.516	1821.275	10733433	87043660
F08	Best	0.004251	6.33E−18	0.001313	0	3.15E−05	0	3.86E−05	3.32E−05	0.0041223	6.23E−05
	Avg	0.061054	0.235935	0.019892	0.004409	0.81577	0.002764	3.614353	4.144109	54.29335	23.29835
	SD	0.328717	1.239318	0.096123	0.076682	3.722776	0.049351	22.90609	26.65522	3.70E+02	1.90E+02
F09	Best	0.036558	4.79E−06	0.163953	0.007396	0.007396	0.216934	5.4E−09	2.58E−06	0.001519	0.003027
	Avg	0.077149	0.017227	0.165403	0.011287	0.072099	0.221949	0.03844	0.051991	0.099817	0.975236
	SD	0.11422	0.103477	0.03343	0.035243	0.132733	0.044534	0.109336	0.147293	0.095948	1.352761

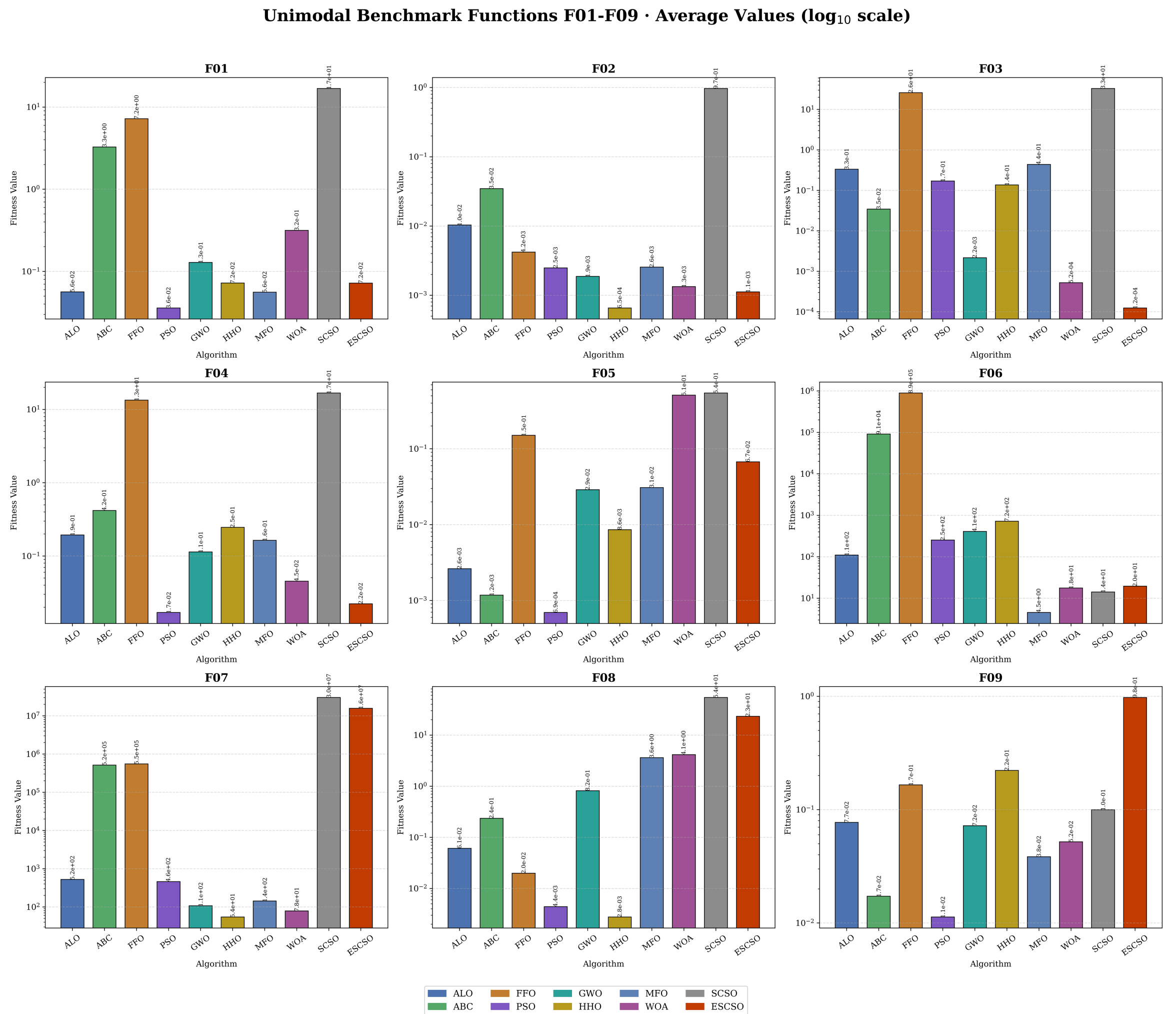

Figure 2: Average fitness values of metaheuristic algorithms vs. the proposed ESCSO algorithm on unimodal benchmark functions F01–F09 (log10 scale and 30 independent runs).

The performance of the proposed ESCSO algorithm is further tested on multimodal functions (i.e., F10–F23) with multiple local minimums. The statistical mean values, best values, and standard deviation for multimodal functions are given in Table 5.

Most of the algorithms, including PSO, GWO, HHO, MFO, WOA, and ESCSO, found global minima with exceptional precision for F10, achieving the optimal best value of 4.44E–16. However, ESCSO's stability was highlighted by its lowest average value of 0.002133 and tiny standard deviation of 0.043123. Like other algorithms, ESCSO also obtained a best value of 0 for F11; however, its consistency was confirmed by the fact that its average and standard deviation values were among the lowest. For more challenging functions, F12 and F13, ESCSO attained near-optimal best values of 0.00000154 and 0.0000266, outperforming most algorithms. While ABC yielded competitive results, other approaches, such as PSO and FFO, produced substantially more variability and less precise solutions. On F14, ESCSO, PSO, GWO, and MFO all reached an optimal best value of zero. Despite this, ESCSO maintained a higher average value of 0.004531 and a lower standard deviation, indicating stability. Also, for F15, ESCSO demonstrated good accuracy with best values close to zero and remained steady across numerous runs, outperforming algorithms such as ABC, FFO, and ALO, which were less stable. On function F16, ESCSO achieved the best optimal value –1.8013, like other algorithms. However, its average value –1.37263 and standard deviation were slightly higher than the leading performers like PSO, suggesting room for improvement. For F17, ESCSO's performance was comparable to other top algorithms, achieving a near-optimal best value and low variability.

Table 5: Best, average, and standard deviation values of algorithms on multimodal functions.

F	Metric	ALO	ABC	FFO	PSO	GWO	HHO	MFO	WOA	SCSO	ESCSO
	Best	0.159196	4.44E−16	0.065066	4.44E−16	4.44E−16	4.44E−16	4.44E−16	4.44E−16	0.00092	4.44E−16
F10	Avg	0.390822	0.125899	0.257288	0.018677	0.031114	0.025417	0.044184	0.005797	1.621075	0.002133
	SD	1.170621	1.088657	0.466961	0.317103	0.391839	0.358304	0.447877	0.083191	3.205971	0.043123
	Best	0.005527	0	0.108046	0	0	4.97479	0	0	0	0
F11	Avg	0.204128	0.113413	0.409092	0.00451	0.883585	4.975052	0.136412	0.01165	0.013445	0.012573
	SD	0.365413	0.896128	0.666802	0.077836	2.10379	0.005282	0.967559	0.187682	0.141567	0.14673
	Best	0.01653	1.38E−13	0.001136	0	6.68E−05	0	3.14E−05	2.3E−06	1.77E−06	1.54E−06
F12	Avg	0.040015	0.227734	0.071279	0.002059	0.500693	0.069487	1.158126	0.998085	0.480409	0.880538
	SD	0.166145	1.311502	0.920119	0.022392	5.323545	0.52277	10.29377	4.970786	1.453215	2.801225
	Best	0.004013	2.55E−05	133.2095	2.55E−05	118.4386	118.4384	118.4395	0.00035	0.012541	2.66E−05
F13	Avg	5.048653	3.666421	206.7866	0.494081	173.9452	119.5307	182.9037	162.2684	45.1963	5.343375
	SD	19.45314	22.70369	38.84715	9.963842	75.48588	17.12911	91.77137	146.2605	19.13457	15.05198
	Best	0.03341	0.007396	4.285565	0	0	0.027125	0	0	0.007524	0
F14	Avg	0.096947	0.047908	4.285565	0.006311	0.064872	0.037679	0.025568	0.039297	1.01651	0.004531
	SD	0.138525	0.380821	8.88E−16	0.060486	0.135867	0.164342	0.152124	0.15274	1.782885	0.031511
	Best	0.000408	1.5E−32	1.35E−06	1.5E−32	4.22E−09	1.5E−32	7E−09	1.65E−08	3.97E−08	4.97E−08
F15	Avg	0.000431	0.008646	0.003709	0.000331	0.002555	0.002442	0.007576	0.021054	0.012122	0.093629
	SD	5.65E−05	8.42E−02	4.51E−02	7.02E−03	9.65E−03	2.66E−02	2.79E−02	7.31E−02	3.21E−01	2.21E−01
	Best	−1.80125	−1.8013	−1.80118	−1.8013	−1.8013	−1.8013	−1	−0.8013	−1.8013	−1.8013
F16	Avg	−1.79281	−1.79613	−1.79281	−1.801	−1.58675	−1.80023	−0.98413	−0.77693	−1.45632	−1.37263
	SD	0.05246	0.05567	0.046294	0.004485	0.272469	0.019659	0.108886	0.101045	0.413979	0.217787
	Best	0.124474	4E−15	0.059041	4.44E−16	0.001061	4.44E−16	0.001043	0.01254	0.001395	0.001935
F17	Avg	0.607226	0.068555	0.414168	0.017376	1.507368	0.022201	2.073082	2.712917	1.598577	1.494604
	SD	0.550107	0.560911	1.288864	0.220238	1.353542	0.26023	1.658184	2.278993	3.559475	2.535742
	Best	0.000271	0	0.039869	0	1.09E-05	0	0.994961	0.000289	3.45E−05	6.45E−08
F18	Avg	0.23517	0.274567	0.174178	0.015583	1.981175	0.000773	2.470305	2.750489	0.015971	0.001826
	SD	0.866978	2.301434	0.537998	0.171678	1.221758	0.007687	1.273882	2.184328	0.149572	0.151599
	Best	0.002088	1.25E−08	0.000708	4.19E−113	1.41E−256	6.11E−16	7.2E−282	7.946E−320	1.3E−05	2.3E−05
F19	Avg	0.00897	0.009705	0.003928	0.000909	0.006024	0.000199	0.000926	0.00052	0.151999	0.170222
	SD	0.023357	0.07854	0.034122	0.010202	0.065071	0.003447	0.015434	0.014076	0.112334	0.307232
	Best	0.020415	0	0.030075	0	0.000108	0	7.91E-05	0.001612	0.001436	0
F20	Avg	0.034572	0.004068	0.047978	0.00073	0.201603	0.001275	0.306414	0.392454	0.259809	0.001916
	SD	0.024703	0.032425	0.062606	0.009567	0.198641	0.021867	0.312795	0.390573	0.407559	0.144364
	Best	−0.5	−0.5584	−0.55429	−0.56126	−0.56044	−0.56126	−0.56126	−0.56126	−0.30141	−0.30141
F21	Avg	−0.49946	−0.11241	−0.45844	−0.55723	0.059454	−0.5594	5.749343	3.175803	0.212222	0.28887
	SD	0.01378	1.570243	0.211813	0.055489	0.089998	0.022606	67.41545	46.14077	0.111232	0.147261
	Best	0	1	0	0	0	0	0	0	0	0
F22	Avg	0.248	1.264735	67.93706	0.08	0.386	0.007	0.395	0.099	33.316	0.016
	SD	1.690117	0.914335	58.85599	1.869117	9.015154	0.113802	10.09183	2.158054	1.8032	0.5153
	Best	−40	−35.785	−40	−40	−40	−40	−28.1404	−19.4469	−40	−40
F23	Avg	−40	−34.4579	−39.969	−39.9861	−40	−40	−34.1326	−31.2133	−19.8615	−39.9692
	SD	7.11E−15	2.05E+00	9.82E−01	4.40E−01	7.11E−15	7.11E−15	7.35E+00	9.51E+00	0.171912	0.972109

Function F18 highlighted ESCSO's ability to produce extremely accurate solutions, with a best value of 6.45E–08, outperforming most algorithms. Similarly, on F19, ESCSO demonstrated exceptional precision, achieving a best value close to the optimal, while maintaining competitive average and standard deviation values. For F20, ESCSO matched the best performing algorithms by achieving an optimal best value of 0. However, its average and standard deviation values indicated slight variability as compared to PSO. In F21, ESCSO's best value −0.30141 was not optimal, but it performed reliably overall, whereas GWO excelled. For functions F22 and F23, ESCSO excelled in terms of convergence along with GWO and ALO. Fig. 3 reports average fitness values for ten algorithms across multimodal benchmark functions F10 to F23 over 30 trial runs. Lower values indicate better performance. Multimodal functions are harder than unimodal ones as the search space contains more local optima, and algorithms that rely too heavily on exploitation tend to stagnate early. ESCSO tends to finish near the front on functions where the algorithms spread apart and stay within the leading group on functions where the differences are small. Across all multimodal functions with varying landscape complexity, that consistency points to balanced search behavior rather than a narrow specialization that performs well in one regime and poorly in another. Its consistent dominance over traditional algorithms like FFO, PSO, and ABC affirms ESCSO's strong exploration and exploitation balance and can be seen in Fig. 3.

The results presented in Table 6. provide a detailed comparison of the performance of various algorithms, including ALO, ABC, FFO, PSO, GWO, HHO, MFO, WOA, SCSO, and ESCSO, across benchmark functions F24 to F30.

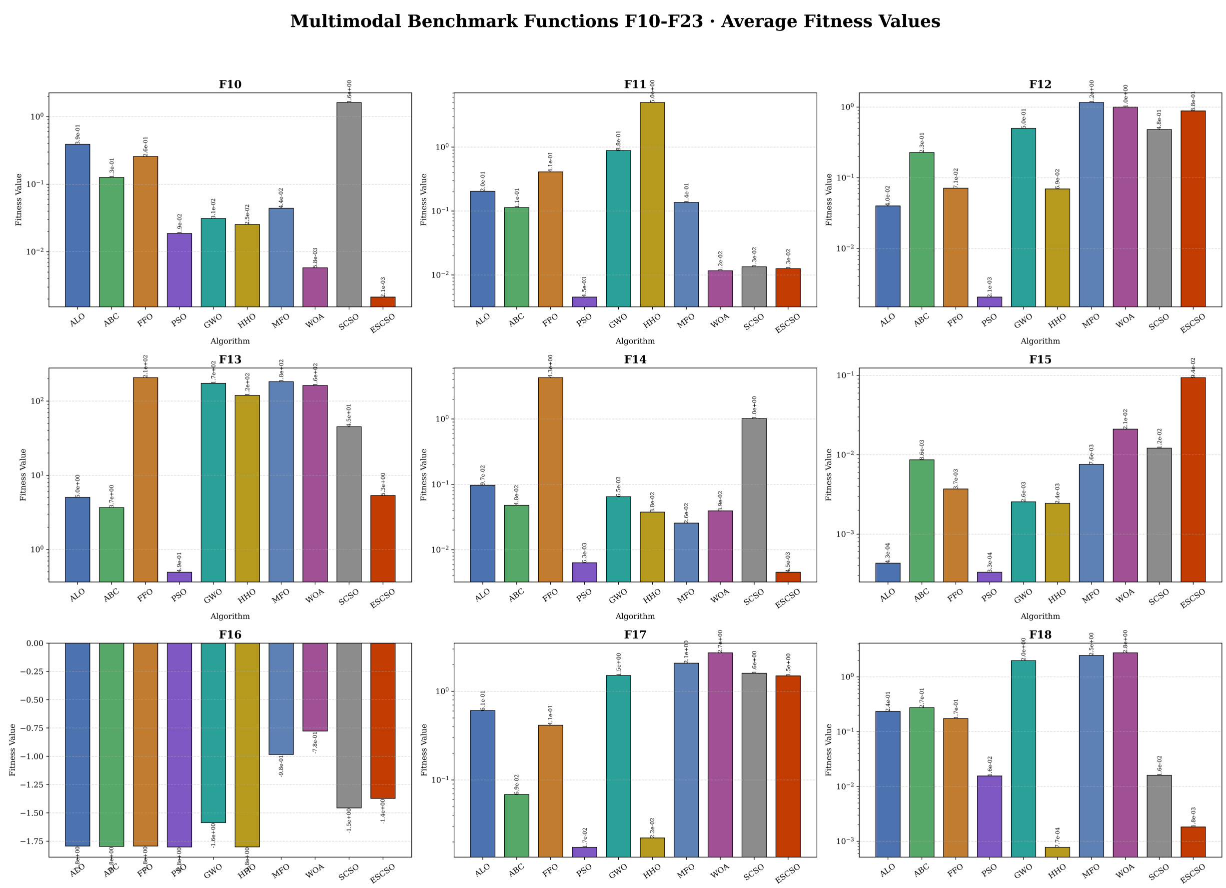

Figure 3: (Continued)

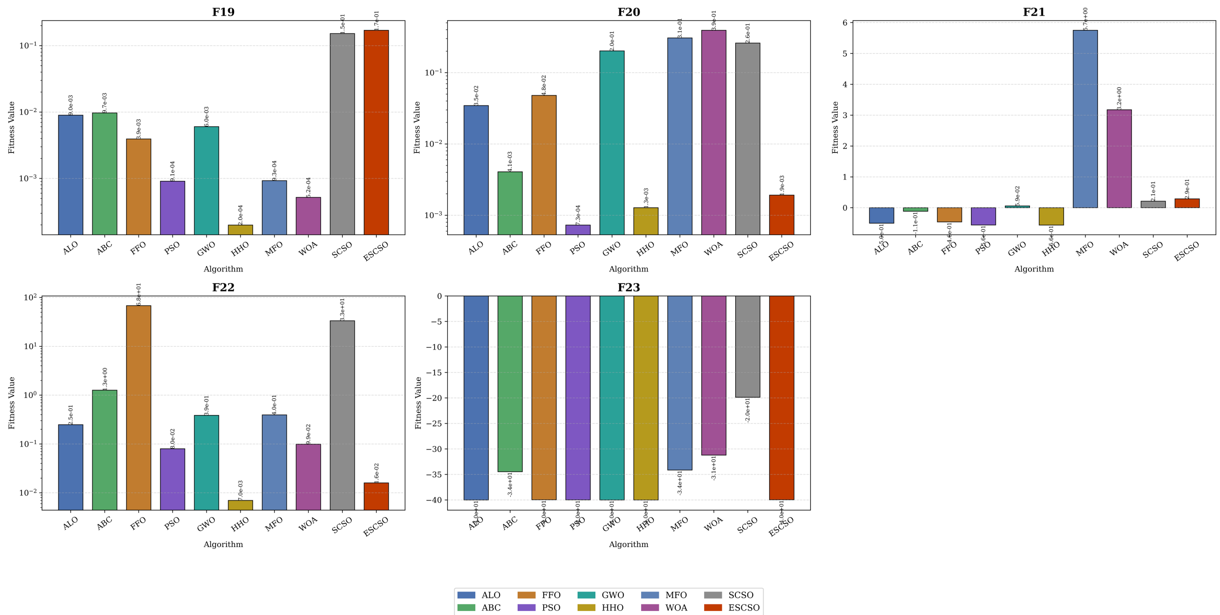

Figure 3: Average fitness values of metaheuristic algorithms vs. the proposed ESCSO algorithm on multimodal benchmark functions F10–F23 (log10 scale and 30 independent runs).

Table 6: Best, average, and standard deviation values of algorithms on hybrid functions.

F	Metric	ALO	ABC	FFO	PSO	GWO	HHO	MFO	WOA	SCSO	ESCSO
F24	Best	4.95E−02	4.44E−16	0.051593	4.44E−16	0.000249	4.44E−16	0.000117	2.579928	0.013316	0.011207
	Avg	7.79E−01	0.082633	0.453208	0.022374	0.300781	0.064474	0.576304	2.033412	4.051337	2.073531
	SD	7.41E−01	0.720789	1.327931	0.345844	0.587714	0.506073	0.832952	1.051712	0.091377	1.957585
F25	Best	1.021584	1	1.038714	1	1	1	1	1	1	1
	Avg	1.299289	1.590847	1.481416	1.044955	1.202814	1.00731	1.067833	1.103576	0.091514	1.680112
	SD	0.919084	5.447243	2.380708	0.667814	2.017578	0.075069	1.266497	2.149547	0.031519	0.054572
F26	Best	9.14E−06	3.02E−09	1.51E−05	0	6.30E−09	1.23E−32	4.13E−05	3.08E−05	3.19E−07	3.33E−07
	Avg	1.80E−04	0.001751	6.45E−04	5.83E−05	0.003502	5.89E−05	0.007494	0.015154	0.036717	0.035857
	SD	0.001914	0.016529	0.005187	5.52E−04	0.00888	0.001254	0.017016	0.035938	0.001344	0.080563
F27	Best	0.023939	2.55E−05	244.9533	2.55E−05	118.4385	118.4384	335.5782	513.3384	1.51E−05	2.93E−05
	Avg	4.994298	2.933299	290.3082	0.00073	167.9646	118.6247	217.6502	331.5908	3.779551	5.798908
	SD	20.25821	16.20317	32.25487	0.006558	62.03399	4.312721	105.1026	154.6442	14.597944	24.62487
F28	Best	0.019341	1.58E−23	0.096912	3.65E−113	2.2E−232	2.28E−47	3.22E−252	9.22E−289	0.001317	0.001772
	Avg	0.173575	1.117985	41.74814	0.184383	0.207902	0.084399	0.316649	0.158917	41.32541	76.85427
	SD	3.286545	10.2392	23.70458	4.499111	3.841319	1.622645	9.217321	4.347483	111.6745	191.1263
F29	Best	0.233804	3.01E−41	0.04747	1.03E−112	7.3E−235	1.25E−52	5.32E−272	1.18E−284	0.007739	0.099391
	Avg	1.282117	4.497069	2.93242	0.006498	2.492262	0.011587	0.085586	0.060378	1099.312	1202.755
	SD	13.91806	11.89909	3.635669	0.111255	53.52692	0.185025	2.183392	1.498352	13727.79	17258.98
F30	Best	837.9263	849.6959	3059.06	837.8111	837.8111	837.8111	837.8111	837.8111	838.5455	837.0001
	Avg	840.0541	989.3156	3059.06	839.1387	849.9315	839.4646	838.4672	838.1326	1689.115	1484.429
	SD	10.54987	169.5475	9.09E−13	21.96372	331.3042	25.92323	18.28716	9.617549	2301.145	2102.247

For Function F24, ABC, PSO, and HHO attained optimal best values of 4.44E−16. Although ESCSO demonstrated competitive performance on F24, it showed a marginally higher best value as compared to the leading algorithms. For function F25, all algorithms, except ALO and FFO, obtained an optimal value of 1. For function F26, PSO, ABC, HHO, and others attained outstanding results with near-zero values. Although ESCSO demonstrated competitive optimal values, its average and standard deviation were marginally high, indicating difficulties in attaining consistent results during multiple trials. GWO exhibited an optimal performance for function F26. The complex multimodal characteristics of F27 showed ABC, and PSO exhibited superior performance as compared to the other algorithms. The hybrid functions; F28 and F29 posed considerable difficulties in convergence to global minima when tested by all the algorithms other than PSO, GWO, HHO, and MFO. For F28 and F29, ESCSO's best values were less than ideal, with notably greater averages and variability. The results highlight the importance of parameter tuning in the proposed ESCSO to effectively balance exploration and exploitation. For F30, ESCSO achieved the lowest best value (837.0001), while PSO, GWO, HHO, MFO and WOA reached 837.8111 and SCSO reached 838.5455. The results demonstrate the superior performance of PSO across most functions, exhibiting excellent best values, lower averages, and minimal variability. Although ESCSO exhibited competitiveness, it encountered challenges with increased variability, especially on hybrid functions. Fig. 4 illustrates the convergence of optimization algorithms based on their average fitness values over 30 trial runs for functions F24 to F30. ESCSO maintains competitive performance, achieving top ranks in several challenging functions, particularly in high-complexity and precision-demanding cases. This further validates the adaptability and robustness of ESCSO across diverse problem types, including hybrid benchmark functions.

Figure 4: Average fitness values of metaheuristic algorithms vs. the proposed ESCSO algorithm on hybrid benchmark functions F24–F30 (log10 scale and 30 independent runs).

The comparative analysis of Tables 4–6 reveals significant insights into the strengths and weaknesses of the algorithms across unimodal, multimodal, and hybrid functions. When looking at the unimodal functions in Table 4, which have a single global optimum, algorithms like PSO and the proposed ESCSO did obtain some of the best results. When looking at Table 5, the multimodal functions GWO, HHO, and MFO were the best at consistently finding global optima. ESCSO, on the other hand, was good at exploring but sometimes had higher standard deviations. The most complex functions to solve were the hybrid functions shown in Table 6. PSO stayed on top by getting the best results across most functions, while ESCSO got similar results but showed more variation, which indicates how sensitive it is to the hybrid functions. Overall, the study demonstrates that different functions can effectively utilize the proposed ESCSO algorithm.

4.5 Statistical Significance: Friedman Test

The Friedman test is a non-parametric statistical method used to compare the performance of multiple algorithms. It ranks the algorithms for each function separately (1 for the best, up to k for the worst) and assesses whether the observed rank differences are statistically significant. For the analysis, the average values of ten algorithms used in this study were ranked on thirty benchmark functions (F01-F30). The resulting rank sums and mean rank values are given in Table 7.

Table 7: Friedman test results ranks of algorithms on 30 benchmark functions.

Algorithm	Sum of Ranks	Mean Rank	Overall Rank
PSO	72	2.40	1
HHO	102	3.40	2
ESCSO	158	5.27	3
ALO	162	5.40	4
ABC	177	5.90	5
GWO	178	5.93	6
WOA	178	5.93	7
MFO	189	6.30	8
FFO	215	7.17	9
SCSO	217	7.23	10

The Friedman test confirmed statistically significant performance differences among the algorithms. PSO and HHO led with mean ranks of 2.40 and 3.40, respectively, clearly separating themselves from the rest and showing steady convergence in all functions. ESCSO obtained third place with a mean rank of 5.27, followed by ALO, ABC, GWO, WOA, and MFO. This reflects solid overall performance for ESCSO, although its edge narrows in the hybrid functions, where its variability increases. FFO and SCSO occupied the bottom ranks owing to higher errors and inconsistent results. Applying the Nemenyi post-hoc critical difference (CD = 2.47 mean-rank points), the gap between PSO and ESCSO (2.87) is statistically significant, whereas ESCSO is not significantly separated from ALO, ABC, GWO, WOA, or MFO. Notably, ESCSO has achieved an advantage over its predecessor SCSO and rose from tenth to third, all owing to the successful integration of selective OBL and the PSO-inspired velocity update mechanisms.

5 Application of ESCSO on Engineering Problems

Seeing the importance of optimization algorithms in the industry, this paper further applied the proposed ESCSO algorithm on two real engineering problems, i.e., Reflectarray Antenna and Microgrid Energy Management, which are discussed in the sub-sections.

5.1 Reflectarray Antenna

Reflect array technology integrates the best properties of parabolic reflectors and phased arrays in a single, flat design for an antenna. It uses different kinds of reflective parts, like microstrip patches, that change the phase of the reflected waves to focus on electromagnetic energy. The main design goal is to minimize the phase error across all reflective elements, subject to design constraints such as allowable reflection coefficients and element spacing [33]. The optimization function for the Reflectarray antenna is defined as:

$$\min f(x) = \sum_{i=1}^{n} \sin^2(x_i) + \prod_{i=1}^{n} \cos^2(x_i) \tag{16}$$

where,

- $x = [x_1, x_2, x_3, \ldots, x_n]$ represents the set of input variables,
- $\sum_{i=1}^{n} \sin^2(x_i)$ represents the phase errors across all elements in the reflectarray,
- $\prod_{i=1}^{n} \cos^2(x_i)$ represents the constructive interference contributing to beam focusing.

As can be observed from Fig. 5, the convergence graph for the Reflectarray antenna optimization using ESCSO demonstrates an effective convergence to global minima. In the initial phase (0–100 iterations), significant fluctuations in the fitness value indicate robust exploration of the search space, with values peaking above 1.0025. Around 100th iteration, the algorithm achieves convergence, stabilizing at a fitness value of approximately 1.0000 and maintaining this stability for the remainder of the iterations (100–500). This highlights ESCSO's ability to balance exploration (searching feasible design space) and exploitation (refining beam focus).

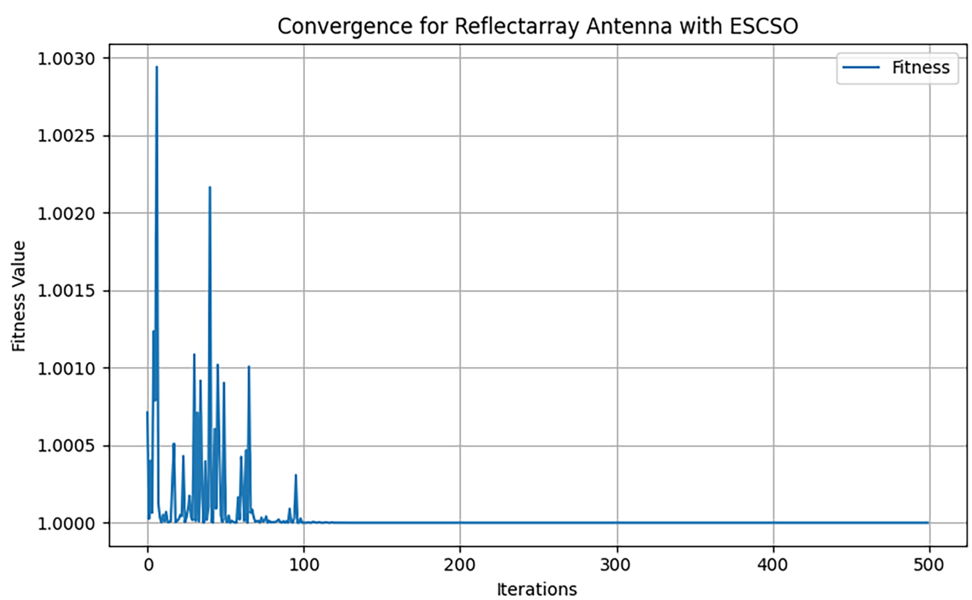

Figure 5: ESCSO performance for the Reflectarray antenna optimization.

5.2 Microgrid Energy Management

Microgrid energy management refers to the process of optimizing the production, storage, and consumption of energy in a microgrid system. A microgrid is a localized energy system that integrates renewable energy sources, traditional energy sources, and energy storage systems to supply electricity to a defined area, such as a community, industrial site, or campus. This problem addresses the operational dispatch (scheduling) sub-problem of microgrid energy management. It selects operating controllable units in such a way that supply meets demand at a minimum cost and penalty. It does not deal with larger problems, such as; capacity planning or storage size [34]. The optimization function for microgrid energy management is defined as:

$$\min f(x) = \sum_{i=1}^{n} \left[(x_i - 2)^2 + |\sin(x_i)| \right] \tag{17}$$

Here, each variable x_i represents the operating set-point of one controllable unit; the quadratic term penalizes deviation from that unit's preferred (economic) operating point, and the $|\sin(x_i)|$ term models periodic, nonlinear inefficiencies in the system. The objective value plotted in Fig. 6 is the value of Eq. (17).

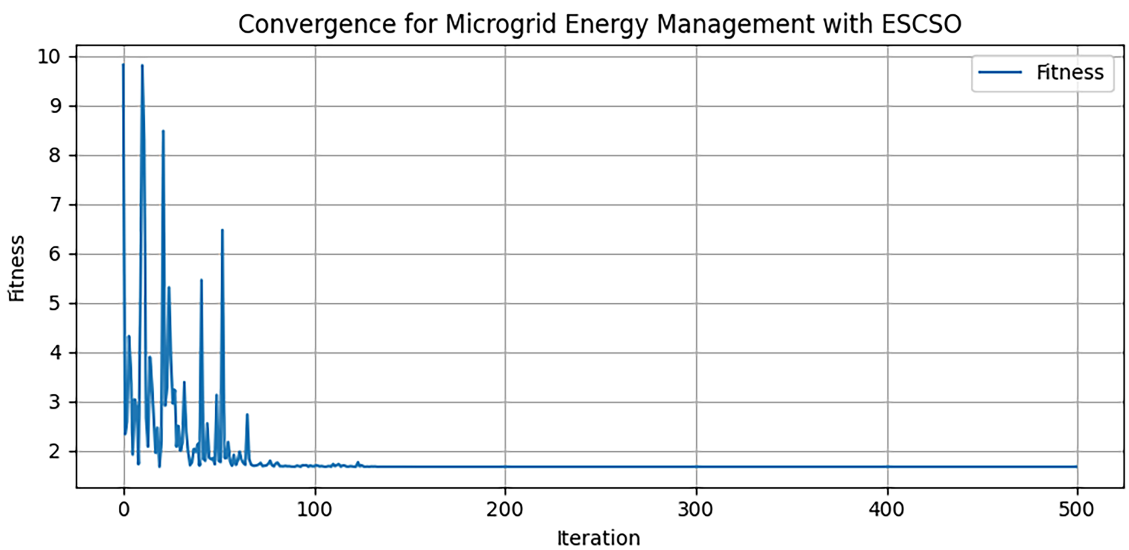

Figure 6: ESCSO Performance for the microgrid energy management.

ESCSO converged effectively on this microgrid dispatch proxy problem, as shown in Fig. 6. During the early iterations (0–50) the objective value fluctuates between roughly 2 and 10: this is the exploration phase, in which selective opposition repositions Sigma-variant cats across the search space and the velocity update has not yet aligned the swarm with the elite Sigma cat. By approximately the 50th iteration the objective value stabilizes near 2.0, indicating convergence to the preferred operating region (the quadratic term is minimized when each x_i approaches its preferred value of 2). From iteration 50 to 500 the value remains steady, reflecting the exploitation phase, in which $|R| \leq 1$ confines the search around the elite. We note that Eq. (17) is a simplified, illustrative objective: it is intended as a proof-of-concept demonstration of ESCSO on an energy-management-motivated landscape and does not yet include realistic operating constraints such as power balance, generator limits, or storage state-of-charge dynamics. Extending ESCSO to a fully constrained microgrid dispatch model, and comparing it against domain-specific optimization methods, is identified as future work in Section 6.

6 Conclusion and Future Works

This paper introduced the Enhanced Sand Cat with Selective Opposition (ESCSO) algorithm, demonstrating significant improvements in exploration-exploitation balance and convergence rate through the selective application of OBL guided by Spearman correlation. The proposed ESCSO algorithm was subjected to a rigorous evaluation across 30 complex optimization functions, and two engineering problems. The comparative analysis showed that ESCSO on the unimodal and most multimodal functions and on both engineering cases, but its variability rises on the hybrid functions, where PSO and HHO remain stronger. The clearest gain is over its predecessor: ESCSO jumped from last place to third in the overall ranking, which indicates that selective opposition strengthens SCSO. A Friedman test across all 30 functions confirmed that the differences between the ten algorithms are statistically significant. Here PSO and HHO are ranked first and second, and ESCSO is ranked third. In the Future, we will explore the adaptability of the ESCSO to different engineering problems promoting continuous progress in the field of optimization algorithms.

Acknowledgement: The authors would like to thank the Research Management Office, Sohar University for supporting this research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Aisha Tanveer and Noraini Ibrahim; methodology, Aisha Tanveer, Muhammad Zubair Rehman, and Noraini Ibrahim; software, Aisha Tanveer; validation, Aisha Tanveer and Noraini Ibrahim; formal analysis, Aisha Tanveer and Noraini Ibrahim; investigation, Aisha Tanveer; resources, Aisha Tanveer and Noraini Ibrahim; data curation, Aisha Tanveer; writing—original draft preparation, Aisha Tanveer; writing—review and editing, Aisha Tanveer, Abdullah Khan, Noraini Ibrahim, Nazri Mohd. Nawi, and Muhammad Zubair Rehman; visualization, Aisha Tanveer; supervision, Noraini Ibrahim and Nazri Mohd. Nawi; project administration, Noraini Ibrahim; funding acquisition, Noraini Ibrahim. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All relevant benchmark functions used are mentioned in tabular format within the manuscript.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A. A survey on new generation metaheuristic algorithms. Comput Ind Eng. 2019;137:106040. doi:10.1016/j.cie.2019.106040.
2. Ezugwu AE, Shukla AK, Nath R, Akinyelu AA, Agushaka JO, Chiroma H, et al. Metaheuristics: A comprehensive overview and classification along with bibliometric analysis. Artif Intell Rev. 2021;54:4237–4316.
3. Rajwar K, Deep K, Das S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artif Intell Rev. 2023;56:13187–257. doi:10.1007/s10462-023-10470-y.
4. Shaikh MS, Lin H, Xie S, Dong X, Lin Y, Shiva CK, et al. An intelligent hybrid grey wolf-particle swarm optimizer for optimization in complex engineering design problem. Sci Rep. 2025;15:18313. doi:10.1038/s41598-025-02154-0.
5. Ahmad MF, Mat Isa NA, Lim WH, Ang KM. Differential evolution: a recent review based on state-of-the-art works. Alex Eng J. 2022;61(5):3831–72.
6. Shojaee Ghandeshtani K, Mashhadi HR. An entropy-based self-adaptive simulated annealing. Eng Comput. 2021;37(2):1329–55. doi:10.1007/s00366-019-00887-x.
7. Guo W, Chen M, Wang L, Mao Y, Wu Q. A survey of biogeography-based optimization. Neural Comput Appl. 2017;28(8):1909–26. doi:10.1007/s00521-016-2179-x.

8. Seyyedabbasi A, Kiani F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Eng Comput. 2022;39(4):2627–51.
9. Peng H, Zhang X, Li Y, Qi J, Kan Z, Meng H, et al. A modified sand cat swarm optimization algorithm based on multi-strategy fusion and its application in engineering problems. Mathematics. 2024;12(14):2153. doi:10.3390/math12142153.
10. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I. Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell. 2018;48(4):805–20. doi:10.1007/s10489-017-1019-8.
11. Tang K, Meng C. Particle swarm optimization algorithm using velocity pausing and adaptive strategy. Symmetry. 2024;16(6):661. doi:10.3390/sym16060661.
12. Reyes-Davila E, Haro EH, Casas-Ordaz A, Oliva D, Avalos O. Differential evolution: A survey on their operators and variants. Arch Comput Methods Eng. 2025;32(1):83–112. doi:10.1007/s11831-024-10136-0.
13. Makhadmeh SN, Al-Betar MA, Abu Doush I, Awadallah MA, Kassaymeh S, Mirjalili S, et al. Recent advances in grey wolf optimizer, its versions and applications: Review. IEEE Access. 2024;12(3):22991–3028. doi:10.1109/ACCESS.2023.3304889.
14. Ji J, Song S, Tang C, Gao S, Tang Z, Todo Y. An artificial bee colony algorithm search guided by scale-free networks. Inf Sci. 2019;473(5):142–65. doi:10.1016/j.ins.2018.09.034.
15. Zhao W, Zhang Z, Wang L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell. 2020;87(5):103300. doi:10.1016/j.engappai.2019.103300.
16. Hussien AG, Gharehchopogh FS, Bouaouda A, Kumar S, Hu G. Recent applications and advances of African vultures optimization algorithm. Artif Intell Rev. 2024;57(12):335. doi:10.1007/s10462-024-10981-2.
17. Yang Z, Deng L, Wang Y, Liu J. Aptenodytes forsteri optimization: Algorithm and applications. Knowl Based Syst. 2021;232:107483.
18. Soto R, Crawford B, Gonzalez F, Vega E, Castro C, Paredes F. Solving the manufacturing cell design problem using human behavior-based algorithm supported by autonomous search. IEEE Access. 2019;7:132228–39. doi:10.1109/access.2019.2940012.
19. Ghasemi M, Davoudkhani IF, Akbari E, Rahimnejad A, Ghavidel S, Li L. A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent flow of water-based optimization. Eng Appl Artif Intell. 2020;92(9):103666. doi:10.1016/j.engappai.2020.103666.
20. Phu-ang A. The new technique based on the galaxy based search algorithm for solving the symmetric travelling salesman problem. In: Proceedings of the 2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON); 2018 Feb 25–28; Chiang Rai, Thailand. p. 131–4. doi:10.1109/ECTI-NCON.2018.8378296.
21. Turgut OE, Turgut MS, Kirtepe E. A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems. Neural Comput Appl. 2023;35(19):14275–14378. doi:10.1007/s00521-023-08481-5.
22. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: A novel physics-based algorithm. Future Gener Comput Syst. 2019;101:646–79.
23. Dhargupta S, Ghosh M, Mirjalili S, Sarkar R. Selective opposition based grey wolf optimization. Expert Syst Appl. 2020;151(11):113389. doi:10.1016/j.eswa.2020.113389.
24. Mehrotra P, Bhardwaj D. HGWO: A novel hybrid optimization algorithm for energy-aware clustering and routing in WSNs. Eng Res Express. 2025;7(3):0352D9. doi:10.1088/2631-8695/ae091b.
25. Subburathinam K, Bakthavatchalam V, Pandian RKC, Subramaniam KM. Enhancing wireless sensor network connectivity and coverage using Hybrid GWO-HSA algorithm. Int J Commun Syst. 2024;37(14):e5858. doi:10.1002/dac.5858.
26. Li Y, Wang G. Sand cat swarm optimization based on stochastic variation with elite collaboration. IEEE Access. 2022;10(4):89989–90003. doi:10.1109/access.2022.3201147.
27. Iraji A, Karimi J, Keawsawasvong S, Nehdi ML. Minimum safety factor evaluation of slopes using hybrid chaotic sand cat and pattern search approach. Sustainability. 2022;14(13):8097. doi:10.3390/su14138097.

28. Jovanovic D, Marjanovic M, Antonijevic M, Zivkovic M, Budimirovic N, Bacanin N. Feature selection by improved sand cat swarm optimizer for intrusion detection. In: Proceedings of the 2022 International Conference on Artificial Intelligence in Everything (AIE); 2022 Aug 2–4; Lefkosa, Cyprus. p. 685–90.
29. Wu D, Rao H, Wen C, Jia H, Liu Q, Abualigah L. Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems. Mathematics. 2022;10(22):4350. doi:10.3390/math10224350.
30. Seyyedabbasi A. Solve the inverse kinematics of robot arms using sand cat swarm optimization (SCSO) algorithm. In: Proceedings of the 2022 International Conference on Theoretical and Applied Computer Science and Engineering (ICTASCE); 2022 Sep 29–Oct 1; Ankara, Turkey. p. 127–31.
31. Kiani F, Anka FA, Erenel F. PSCSO: Enhanced sand cat swarm optimization inspired by the political system to solve complex problems. Adv Eng Softw. 2023;178:103423.
32. Seyyedabbasi A. A reinforcement learning-based metaheuristic algorithm for solving global optimization problems. Adv Eng Softw. 2023;178(3):103411. doi:10.1016/j.advengsoft.2023.103411.
33. Qureshi SA, Kamarudin MR, Abbasi MI, Yamada Y, Dahri MH, Abidin ZZ, et al. The 6G reconfigurable reflectarray antenna using a gold-VO^2 bilayer structure. Prog Electromagn Res M. 2024;130:63–70. doi:10.2528/PIERM24073002.
34. Seshu Kumar R, Phani Raghav L, Koteswara Raju D, Singh AR. Intelligent demand side management for optimal energy scheduling of grid connected microgrids. Appl Energy. 2021;285(3):116435. doi:10.1016/j.apenergy.2021.116435.