International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Horseshoe Crab Optimization: A Novel Nature-inspired Metaheuristic based on Spawning and Satellite Male Aggregation Behaviours

Ayman Alqafaan[1] Omar Al Sayyed[2] Belal Batiha[3]
Gulnara Bektemyssova[4] Zeinab Montazeri[5] Mohammad Dehghani[5]*
Om Parkash Malik[6] Kei Eguchi[7]

[1]*Department of Cyber Security, Al Zaytoonah University of Jordan, Amman 11733, Jordan*
[2]*Department of Mathematics, Faculty of Science, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan*
[3]*Department of Mathematics, Faculty of Science, Jadara University, Irbid 21110, Jordan*
[4]*Department of Computer Engineering, International Information Technology University, Almaty 050000, Kazakhstan*
[5]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran*
[6]*Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada*
[7]*Department of Information Electronics, Fukuoka Institute of Technology, Japan*
* Corresponding author's Email: adanbax@gmail.com

Abstract: A novel population-based metaheuristic algorithm, termed Horseshoe Crab Optimization (HCO), inspired by the reproductive behaviour of horseshoe crabs (*Limulidae*) in tidal ecosystems, is introduced in this paper. HCO models two key biological mechanisms: spawning aggregation and satellite male behaviour. Spawning aggregation, occurring during high-tide periods, drives the population toward favourable regions for egg-laying and serves as an effective global exploration mechanism, while satellite male behaviour, in which males position themselves around high-quality females and adjust their locations precisely, is modelled as a targeted local exploitation process. This mechanism facilitates focused search in regions containing high-quality solutions and, through the incorporation of stochastic components, prevents premature convergence to local optima. The mathematical formulation of HCO incorporates tidal-phase modulation, adaptive convergence factors, and stochastic intensities, enabling dynamic transitions between exploration and exploitation phases. Each member of the population represents a candidate solution, and updates are governed by movement rules inspired by reproductive behaviour along with a fitness-based acceptance criterion, ensuring both solution diversity and convergence stability. HCO is evaluated on 23 benchmark functions, including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal problems, and its performance compared with nine state-of-the-art metaheuristic algorithms. The results demonstrate that HCO achieves first-rank performance in 20 out of 23 functions, exhibiting superior convergence speed, solution accuracy, and robustness, thereby establishing it as a reliable and versatile framework for addressing complex, high-dimensional optimization problems.

Keywords: Horseshoe crab, Metaheuristic, Optimization, Exploration, Exploitation, Spawning aggregation, Satellite male.

1. Introduction

Optimization is one of the fundamental and pervasive concepts in science and engineering, serving as the cornerstone for decision-making, design, control, and prediction across countless domains. In its most general form, optimization refers to the process of determining the best solution among a set of feasible alternatives according to a defined objective or fitness function [1]. This process is crucial in engineering design, operations research, artificial intelligence, economics, and data analytics, where it is essential to identify parameter configurations that minimize cost, maximize efficiency, or improve performance within defined constraints [2].

Mathematically, an optimization problem involves finding a vector x^* in a feasible search space S, such that the objective function $f(x)$ reaches its global minimum or maximum. The challenge arises when $f(x)$ is nonlinear, multimodal, non-differentiable, or defined over high-dimensional and complex landscapes where traditional analytical or deterministic methods become inefficient or intractable [3].

To address these challenges, a broad spectrum of optimization strategies has been developed. Traditional deterministic methods, such as gradient-based techniques, linear programming, and convex optimization, rely on strict mathematical formulations and gradient information to systematically converge to a local or global optimum. While being effective for convex, smooth, and well-behaved functions, these methods often fail when applied to non-convex or discontinuous problems, as they can easily become trapped in local optima or require extensive derivative computations [4-6]. In contrast, stochastic approaches, particularly metaheuristic algorithms, provide a flexible and probabilistic mechanism for navigating complex search spaces without relying on gradient information or specific problem structures [7].

Metaheuristic algorithms have emerged as one of the most versatile and powerful tools for solving complex optimization problems. They are high-level search strategies that guide subordinate heuristics to explore and exploit the solution space effectively [8]. These algorithms are characterized by their robustness, generality and adaptability to a wide variety of optimization problems, including continuous, discrete, multi-objective, and constrained [9]. Over the past two decades, metaheuristics have demonstrated remarkable success in addressing real-world optimization challenges in fields such as structural design, power systems, image processing, machine learning, and industrial engineering [10-14].

A key strength of metaheuristic algorithms lies in their simplicity, scalability, and flexibility. Unlike classical optimization methods that require problem-specific adaptations or differentiable functions, metaheuristics operate through generalizable mathematical operators inspired by natural, physical, or social processes. They employ randomization and population-based search to maintain diversity and avoid premature convergence, while adaptive updating rules ensure convergence toward high-quality solutions. Consequently, metaheuristics have become indispensable in modern computational optimization due to their ability to efficiently approximate near-global optimum for complex, multidimensional problems [15].

Two fundamental concepts underlie the design and functioning of all metaheuristic algorithms: exploration and exploitation [16].

- Exploration refers to the capability of an algorithm to broadly sample the search space and investigate new, unvisited regions. This phase prevents the population from becoming trapped in local minima and enhances the algorithm's ability to discover the global optimum. It often relies on stochastic perturbations, random walks, or global search operators that encourage diversity and large-scale movement [17].
- Exploitation, on the other hand, focuses on intensifying the search around promising solutions that have already been identified. Through local refinement, learning mechanisms, and convergence strategies, the algorithm enhances solution precision and stability [18].

Maintaining an effective balance between exploration and exploitation is critical. Excessive exploration can lead to slow convergence and inefficient searches, whereas overemphasis on exploitation increases the risk of premature convergence to suboptimal solutions. Successful metaheuristic design therefore involves dynamically regulating the interplay between these two processes, often through adaptive parameters, probabilistic transitions, or multi-phase update rules that shift emphasis from exploration in early iterations to exploitation in later stages.

Despite their flexibility, metaheuristics are inherently stochastic. Their random nature implies that global optimality cannot be mathematically guaranteed. Instead, they aim to produce high-quality, near-optimal solutions within reasonable computational time [19]. Consequently, continuous innovation in algorithmic design has been motivated by the desire to enhance search efficiency, convergence stability, and robustness. This has led to the development of numerous metaheuristic algorithms [20].

Metaheuristic algorithms have been designed by drawing inspiration from a wide range of sources, including biological and genetic sciences, animal and wildlife behaviors, physical and chemical processes, human social interactions, competitive games, and other systems that inherently exhibit optimization-like dynamics. In general, metaheuristics can be categorized into several principal families based on their inspiration and mechanism [21].

The first major family includes Evolutionary-based algorithms, which are derived from the principles of Darwinian evolution. The Genetic Algorithm (GA) [22] is the most classical representative, simulating natural selection, crossover, and mutation among a population of candidate solutions (chromosomes). Through iterative reproduction and competition, fitter individuals propagate their features to the next generation, achieving an effective combination of exploration (through mutation and recombination) and exploitation (through selection). GA has served as a foundational model for many subsequent evolutionary techniques.

The second family, Swarm-based algorithms, is inspired by collective behavior and self-organization in biological populations. The Particle Swarm Optimization (PSO) [23] algorithm, for instance, mimics the coordinated movement of bird flocks and fish schools, where each particle adjusts its trajectory based on its personal best position and the best position found by the group. This dual influence ensures a dynamic trade-off between global search and local refinement. Similarly, Ant Colony Optimization (ACO) [24] models the pheromone communication system of ants to discover the shortest path between a nest and a food source. As ants deposit pheromones along favorable routes and evaporation removes weaker trails, the system collectively converges toward optimal paths. Another example, Grey Wolf Optimization (GWO) [9], reproduces the social hierarchy and cooperative hunting strategy of grey wolves, dividing the population into alpha, beta, delta, and omega levels. By following and encircling the prey (the optimal solution), GWO effectively balances convergence precision and diversity retention.

A third group, Physics-based algorithms, draws upon fundamental principles of mechanics, thermodynamics, and electromagnetism. Simulated Annealing (SA) [25] models the annealing process in metallurgy, where gradual cooling allows atoms to reach low-energy, stable configurations. At high temperatures, the system explores broadly (exploration), while as temperature decreases, the search becomes more focused (exploitation). Spring Search Algorithm (SSA) [26] simulates the elastic behavior of springs, where agents adjust their positions under attractive and repulsive forces until equilibrium is reached. Physics-based algorithms often exhibit simplicity in formulation yet maintain strong mathematical stability and convergence behavior.

The fourth category comprises Human-based algorithms, which model learning, teaching, and social interaction processes among humans. For example, Teaching–Learning-Based Optimization (TLBO) [27] simulates classroom dynamics, where the teacher (best solution) improves the knowledge of learners, and learners further enhance their understanding through peer interaction. Likewise, the Barber Optimization Algorithm (BOA) [28] is inspired by service dynamics between barbers and customers, representing iterative refinement through feedback and mutual adaptation. These algorithms often excel in parameter-free designs and reflect rational human decision-making mechanisms.

The Game-based algorithms take inspiration from rules and strategies of competitive games. Darts Game-Based Optimization (DGO) [29] models the process of throwing darts toward a target, combining accuracy and randomness to refine search directions. Golf Optimization Algorithm (GOA) [30] simulates how golfers iteratively adjust their shots to minimize the distance to the hole, reflecting adaptive learning through trial and feedback. Such algorithms integrate notions of competition, learning from failure, and incremental improvement, effectively mirroring the optimization process itself.

Emergence of a new generation of metaheuristic algorithms in recent years reflects growing interdisciplinary creativity in combining natural, social, and physical inspirations. The Mother Optimization Algorithm (MOA) [31] is a human-based approach that simulates maternal behavior through three core phases—teaching, advising, and nurturing—to gradually shift the population from exploration to exploitation. The "mother" (best solution) directs offspring toward broader random movements during the teaching phase, facilitates learning from past experiences during advising, and refines convergence during nurturing. Experimental analyses on benchmark functions and engineering problems have shown that MOA effectively balances global and local search while outperforming several established algorithms.

The Artificial Orca Optimizer (AOO) [32] models the organized hunting strategies of killer whales. It employs guidance and switching mechanisms to prevent premature convergence toward the center of the search space and maintain population diversity. Agents adaptively switch between strategies based on environmental feedback, ensuring fast convergence and robustness against local optima.

The Raindrop Optimizer (RO) [33] draws inspiration from the behavior of raindrops upon impact. Each raindrop acts as an independent agent that disperses, merges, and diffuses upon contact with surfaces, collectively enhancing diversity and intensifying the search around promising regions. This model demonstrates superior performance in engineering and AI-related optimization tasks due to its stable and dynamic search dynamics.

Human-social paradigm is introduced in the Knowledge Propagation Optimization (KPO) [34] algorithm, where agents share and propagate knowledge across a dynamic network. Weighted information exchange allows beneficial experiences to spread faster through the population, thereby accelerating convergence and preventing entrapment in local minima. KPO has shown strong potential in photovoltaic system optimization and parameter tuning tasks.

The Wave Optics Optimizer (WOO) [35] leverages optical wave phenomena—particularly diffraction and interference—to regulate search behavior. Each candidate solution represents a light wave that interacts with others; diffraction enhances exploration, while interference amplifies promising zones. This wave-based interaction maintains both global diversity and focused exploitation.

The Equilibrium Optimizer (EO) [36] is based on mass balance principles in control volume systems. Each particle possesses a concentration level and updates its position relative to equilibrium candidates through a randomly generated rate of production. The process maintains population diversity and avoids premature stagnation, achieving reliable performance across a variety of engineering problems.

The Chimp Optimization Algorithm (ChOA) [37] models the cooperative and competitive hunting behavior of chimpanzees. By incorporating leadership hierarchies and stochastic movements, it balances social coordination with random exploration, effectively enhancing convergence precision and population diversity.

Similarly, the Spotted Hyena Optimizer (SHO) [38] captures the multi-phase hunting behavior of spotted hyenas—searching, encircling, and attacking. This structure enables gradual transition from broad exploration to local exploitation, ensuring steady and accurate convergence.

The Orca Predation Algorithm (OPA) [39] further extends marine behavioral modeling by representing orca hunting dynamics through driving, encircling, and attacking phases with weighted transitions. Agents are influenced both by elite and random individuals, maintaining diversity and adaptive focus.

The Moth–Flame Optimization (MFO) [40] algorithm simulates the transverse orientation navigation of moths relative to light sources. Candidate solutions (moths) spiral toward superior positions (flames), dynamically shifting from wide exploration to tight exploitation. This approach achieves strong convergence characteristics in complex, multidimensional search spaces.

Collectively, these algorithms illustrate the remarkable breadth of inspiration underlying metaheuristic design and highlight the continuous evolution of optimization science. Despite their seeming diversity, they all share the same objective: to efficiently explore the search space, exploit promising regions, and achieve near-global optimality. Nevertheless, as the No Free Lunch (NFL) theorem [41] asserts, no single algorithm can outperform all others across every class of optimization problem. Therefore, the ongoing development of innovative metaheuristic algorithms remains essential to address emerging complex and multidimensional optimization challenges in modern science and engineering.

To the best of the authors' knowledge, no metaheuristic algorithm has previously been developed based on the unique reproductive and aggregation behaviors of horseshoe crabs. These ancient marine arthropods exhibit remarkable collective patterns, including seasonal beach spawning and satellite male aggregation, which inherently balance global exploration and local exploitation in natural environments. Inspired by their behavior and motivated by the NFL theorem, a novel optimization algorithm, termed Horseshoe Crab Optimization (HCO), that abstracts these biological strategies into a computational framework for solving complex optimization problems is prop.

In HCO, the exploration phase emulates seasonal spawning, where individuals collectively move toward favorable "beach" locations (represented by the best current solutions) modulated by a tidal coefficient. This mechanism ensures wide-area sampling of the search space, promoting discovery of promising regions while maintaining population diversity. The exploitation phase models the satellite male behavior, in which individuals converge toward selected high-quality solutions while small random perturbations allow fine-tuned local search and prevent premature convergence. Both phases are rigorously formalized through mathematical operators that capture the stochastic and adaptive nature of horseshoe crab movements, including time-dependent tidal effects, distance-based attraction, and controlled random perturbations.

The HCO algorithm offers several key advantages: a biologically grounded multi-phase search process, dynamic balance between global exploration and local refinement, natural adaptability

Received: October 2, 2025. Revised: November 15, 2025.

to maintain diversity, and precise convergence toward optimal solutions through distance-aware, diminishing local adjustments. By translating the ecological behaviors of horseshoe crabs into mathematically precise operators, HCO bridges the gap between biological observation and computational optimization, providing a robust and general-purpose optimization framework.

The main contributions of this study are summarized below:

- Development of the HCO algorithm, inspired by the seasonal spawning and satellite male aggregation behaviors of horseshoe crabs.
- Formal mathematical modeling of exploration and exploitation phases based on tidal modulation, directed attraction to superior individuals, and adaptive stochastic perturbations.
- Comprehensive evaluation on a suite of benchmark functions, covering unimodal, high-dimensional multimodal, and fixed-dimensional multimodal landscapes.
- Comparative analysis with state-of-the-art metaheuristic algorithms to demonstrate competitiveness, robustness, and convergence efficiency.

The remainder of this paper is structured as follows: the biological inspiration and detailed mathematical formulation of HCO are presented in Section 2. Simulation studies, statistical performance results, and comparative analysis are provided in Section 3. Finally, conclusions of the study and potential avenues for future research are given in Section 4.

2. Introducing and mathematical modelling of horseshoe crab optimization (HCO)

Theoretical foundation and main inspiration behind the proposed *Horseshoe Crab Optimization* algorithm are presented in this section. Inspired by the natural and reproductive behaviour of horseshoe crabs in tidal ecosystems, HCO integrates the dynamics of environmental adaptation, cyclic aggregation, and cooperative mating to simulate a balance between exploration and exploitation. To employ this biologically inspired strategy in optimization problems, the behavioural mechanisms are mathematically modelled to enable efficient global and local search capabilities.

2.1 Biological inspiration and behavioural analysis

Horseshoe crabs (*Limulidae*) are ancient marine arthropods that have existed for more than 400 million years. Their life cycle, adaptive behavior, and interaction with the coastal environment make them a unique biological system for computational modeling. The most characteristic behaviors relevant to optimization include tidal-based spawning aggregations and satellite mating behavior.

During specific periods of the lunar and tidal cycle, large numbers of horseshoe crabs migrate from deeper waters to sandy beaches to spawn. This spawning aggregation occurs mostly during *high tide*, when the conditions are favorable for egg deposition. Individuals use environmental cues such as water temperature, moonlight, and tidal phase to synchronize their movement. This coordinated behavior reflects an effective *exploration mechanism*, where multiple individuals simultaneously spread out and explore new regions (the beaches) in search of optimal spawning sites. Hence, this collective migration is analogous to the global exploration phase in optimization, allowing the population to investigate vast regions of the search space.

Once the spawning females bury their eggs in the sand, multiple males gather around each female. These *satellite males* compete to fertilize the eggs, maintaining proximity to the female while adjusting their position precisely. This cooperative yet competitive behavior emphasizes localized searching near promising targets. Therefore, this pattern represents the exploitation phase of the optimization process, focusing on intensifying the search around high-quality solutions (females) while allowing minor stochastic adjustments to prevent stagnation.

In summary:

- **Spawning aggregation** ⟶ *Exploration phase*: global search inspired by synchronized movement driven by tidal cycles.
- **Satellite male aggregation** ⟶ *Exploitation phase*: local refinement inspired by male convergence around high-quality female solutions.

2.2 Mathematical formulation and population initialization

To mathematically simulate the population of horseshoe crabs, each individual is represented as a potential solution vector in an m-dimensional search space. The initial population matrix X of size $N \times m$ is expressed as:

 DOI: 10.22266/ijies2026.0131.22

Received: October 2, 2025. Revised: November 15, 2025.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (1)$$

Each element $x_{i,d}$ represents the position of the i-th crab in the d-th dimension, generated using:

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \quad (2)$$

where:

- lb_d and ub_d denote the lower and upper bounds of the search space in dimension d,
- $r \sim U(0,1)$ is a uniformly distributed random number controlling stochastic initialization.

The fitness of each individual is then evaluated as:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (3)$$

where $F(X_i)$ represents the objective function value corresponding to the position X_i. Initialization ensures that the crabs (solutions) are uniformly distributed across the search space to maximize exploration in early iterations.

2.3 Exploration phase: Spawning aggregation behaviour

In nature, horseshoe crabs gather on coastal beaches for spawning during specific tidal and seasonal periods. This phenomenon, known as Spawning Aggregation, is one of the most remarkable collective behaviors of this species. The temporal pattern of aggregation is directly influenced by the tidal cycle and environmental conditions. When the water level rises (high tide), the crabs migrate toward the shore and exhibit intense activity searching for suitable spawning sites. In contrast, during low tide, the movement and exploration activities diminish.

This behavior aligns closely with the concept of global exploration in metaheuristic algorithms, where the population disperses widely in the search space to identify new and potentially optimal regions.

In the proposed HCO formulation, the position of the current best individual is interpreted as an attractive "beach" or spawning site toward which other crabs tend to move during favourable tidal phases. This modelling choice captures the biological observation that many individuals converge to the same preferred coastal locales when environmental cues indicate a suitable spawning window. The multiplicative tidal coefficient $M(t)$ modulates the strength of this attraction over time, amplifying movement toward the beach during high-tide phases and attenuating it during low-tide phases. Complementing this directed attraction, the additive stochastic term $\alpha(t) \cdot r \cdot (X_R - X_i)$ represents opportunistic, random displacements—modeling exploratory detours, environmental perturbations, and the propensity of individuals to sample new regions. As iterations progress, $\alpha(t)$ decays, reducing random excursions and allowing the population to transition from broad exploration toward more focused search behavior around promising areas.

The mathematical model of this behavior in HCO is defined as:

$$X_i^{P1} = X_i + M(t) \cdot r \cdot (X_{best} - I \cdot X_i) + \alpha(t) \cdot r \cdot (X_R - X_i) \quad (4)$$

where the tidal function $M(t)$ is formulated as:

$$M(t) = 1 + \sin(2\pi \frac{t}{T_{tidal}}) \quad (5)$$

and the stochastic intensity term is given by:

$$\alpha(t) = \left(\frac{1}{t+1}\right)^2 \quad (6)$$

In these equations:

- $M(t)$ is a periodic function oscillating between 0 and 1, modeling the tidal phase variation.
  When $M(t) \approx 1$ (the high tide period), exploration activity intensifies.
- $\alpha(t)$ controls the random movement strength, decreasing as iterations progress.
- X_R represents a random reference position that diversifies movements.
- $I \in \{1,2\}$, $r \sim U(0,1)$ and T_{tidal}=12 (representing months of the year) are constant parameters of the tidal model.

The position is updated only if the new fitness improves the previous one:

 DOI: 10.22266/ijies2026.0131.22

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & else \end{cases} \quad (7)$$

This adaptive process allows the algorithm to oscillate between wide-ranging and focused exploration, reflecting the natural cyclic spawning of horseshoe crabs.

2.4 Exploitation phase: Satellite male aggregation behaviour

In natural systems, once the females lay their eggs, multiple males surround each female as *satellite males*, competing for fertilization. This pattern represents a cooperative and targeted local search process. Each male adjusts its position based on proximity to the female, mirroring a focused optimization strategy near promising solutions.

In HCO, for each member i, individuals with superior fitness values are considered female candidates. The male i randomly selects one female and moves toward her position. A stochastic noise component $\varepsilon(t)$ ensures diversity and prevents premature convergence.
The mathematical formulation is as described below:

$$X_i^{P2} = X_i + \sigma(t) \cdot r \cdot (Female_i - I \cdot X_i) + \varepsilon(t) \quad (8)$$

where:

$$\varepsilon(t) = \frac{K_\varepsilon \cdot (1 - 2r) \cdot |UB - LB|}{t + 1} \quad (9)$$

and

$$\sigma(t) = 1 - \left(\frac{t}{T}\right)^{1+r} \quad (10)$$

In these equations:

- $Female_i$: the selected female position.
- $K_\varepsilon \in \{0,1\}$: random binary parameter determining whether noise is applied.
- $\varepsilon(t)$: small random perturbation decreasing with time.
- $\sigma(t)$: convergence factor that reduces step size as iterations progress.
- $I \in \{1,2\}$, $r \sim U(0,1)$: random variables similar to the exploration phase.

The position update rule follows a fitness-based acceptance criterion:

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & else \end{cases} \quad (11)$$

This mechanism focuses the search around high-quality solutions, allowing the population to converge while maintaining adaptability, analogous to the sustained satellite behaviour of male horseshoe crabs.

2.5 Computational complexity analysis

The computational complexity of the proposed HCO algorithm primarily depends on the initialization process and the iterative updating phases, including exploration and exploitation. In the initialization stage, random generation of the population consisting of N individuals across a D-dimensional search space requires a computational cost of $O(N \times D)$.

During the optimization process, each individual undergoes two successive update mechanisms per iteration: spawning aggregation (exploration) and satellite male aggregation (exploitation). In the exploration phase, the updating process involves random coefficient generation, best-solution referencing, and position adjustment for all population members, resulting in a computational cost of $O(N \times D)$. Similarly, in the exploitation phase, candidate positions are refined using female selection and adaptive perturbation mechanisms, leading to an equivalent complexity of $O(N \times D)$.

Therefore, considering both update phases repeated over T iterations, the total computational cost of the iterative process becomes $O(T \times N \times D)$. Combining this with the initialization step, the overall computational complexity of the HCO algorithm can be expressed as:

$$O(N \times D) + O(T \times N \times D) \approx O(T \times N \times D)$$

Thus, the overall time complexity of HCO grows linearly with the number of iterations, population size, and problem dimensionality, which is consistent with the majority of population-based metaheuristic algorithms.

2.6 Algorithmic procedure and pseudocode

The overall process of HCO is summarized below:

The population of crabs is initialized randomly in the search space. During each iteration, individuals undergo a spawning aggregation (exploration phase) driven by tidal dynamics, followed by satellite male aggregation (exploitation phase) directed toward

superior solutions. This cyclic interaction between exploration and exploitation continues until a termination condition (e.g., maximum iterations or acceptable fitness) is met.
The pseudocode of the proposed algorithm is presented in Algorithm 1.

Algorithm 1: Pseudocode of Horseshoe Crab Optimization

Input: Objective function $F(x)$, population size N, max iterations T
Output: Best solution X_{best} and F_{best}

1: Initialize population X_i using Eq. (2) within LB, UB
2: Evaluate fitness $F(X_i)$ for all individuals
3: Identify the best individual X_{best}
4: For t = 1 to T do
5: For each individual i do
6: // Exploration Phase (Spawning Aggregation)
7: Compute $M(t)$ and $\alpha(t)$ using Eqs. (5) and (6)
8: Generate X_i^{P1} using Eq. (4)
9: If $F(X_i^{P1}) < F(X_i)$, update $X_i \leftarrow X_i^{P1}$
10: End For
11: Update X_{best}
12: For each individual i do
13: // Exploitation Phase (Satellite Male Aggregation)
14: Select $Female_i$ among better individuals
15: Compute $\sigma(t)$ and $\varepsilon(t)$ using Eqs. (9) and (10)
16: Generate X_i^{P2} using Eq. (8)
17: If $F(X_i^{P2}) < F(X_i)$, update $X_i \leftarrow X_i^{P2}$
18: End For
19: Update X_{best}
20: End For
21: Return X_{best} and F_{best}

---

3. Simulation studies, results, and discussion

To rigorously assess the optimization performance and robustness of the proposed HCO algorithm, a comprehensive set of simulation experiments was conducted. This section presents the benchmark-based evaluation, comparative analysis with established metaheuristic algorithms, and an in-depth discussion of the results. The experiments were designed to evaluate the convergence behavior, accuracy, stability, and exploration–exploitation balance of HCO across diverse optimization landscapes.

All algorithms were implemented under identical computational conditions, and statistical indicators were employed to ensure a fair and reliable comparison. The obtained outcome demonstrates the remarkable capability of the HCO algorithm to achieve high-quality solutions consistently across various types of benchmark functions.

3.1 Benchmark functions and experimental setup

The performance of HCO was tested using 23 benchmark functions from three categories [42]:

- **Unimodal functions (F1–F7)** are used to evaluate the exploitation ability of the algorithm in locating the global optimum.
- **High-dimensional multimodal functions (F8–F13)** test the algorithm's capacity to escape local optima and explore complex landscapes in high-dimensional spaces.
- **Fixed-dimensional multimodal functions (F14–F23)** assess the overall exploration-exploitation balance of the algorithm in problems with multiple global and local optima.

The proposed HCO has been compared with nine competing metaheuristics, namely: Archimedes Optimization Algorithm (AOA) [43], Tunicate Search Algorithm (TSA) [16], Whale Optimization Algorithm (WOA) [44], Multi-Verse Optimizer (MVO) [45], Grey Wolf Optimizer (GWO) [9], Teaching–Learning-Based Optimization (TLBO) [27], Gravitational Search Algorithm (GSA) [2], Particle Swarm Optimization (PSO) [23], and Genetic Algorithm (GA) [22].

All algorithms were executed for 1,000 iterations over 20 independent runs for each function. The performance was evaluated using six statistical metrics: mean, best, worst, standard deviation (std), median, and rank. Optimization results are reported in Tables 1–3, while convergence curves are illustrated in Fig. 1.

3.2 Performance analysis on unimodal functions (F1–F7)

The unimodal test functions primarily evaluate the exploitation capability of an optimization algorithm, as they contain a single global optimum without local traps. Therefore, superior performance in these functions indicates strong local search ability and high convergence accuracy.

According to Table 1, the proposed HCO algorithm achieved the best overall performance, obtaining the first rank in six out of seven functions

Table 1 Optimization results of unimodal test functions

F		HCO	AOA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F1	mean	2.3E-193	6.1E-174	1.37E-46	4.9E-150	0.143116	9.33E-59	3.06E-75	9.75E-17	0.804204	31.44179
	best	9.2E-202	3.1E-250	6.74E-50	4.5E-165	0.074906	2.96E-61	6.82E-79	5.14E-17	9.02E-06	15.99706
	worst	3.1E-192	1.2E-172	1.97E-45	8.4E-149	0.269029	6.13E-58	2.95E-74	1.54E-16	15.85564	63.58857
	std	0	0	4.38E-46	1.9E-149	0.046408	1.68E-58	6.49E-75	2.46E-17	3.542842	11.20346
	median	2.4E-196	2.9E-207	5.5E-48	7.7E-162	0.134148	2.55E-59	1.15E-75	1.02E-16	0.001096	27.87951
	rank	1	2	6	3	8	5	4	7	9	10
F2	mean	5.8E-99	1.22E-87	2.5E-29	1.3E-105	0.26341	1.47E-34	1.21E-38	0.081746	0.737989	3.269899
	best	4.5E-102	2.4E-130	1.55E-30	4.5E-110	0.180146	8.54E-36	2.27E-40	3.6E-08	0.235933	1.675419
	worst	5.62E-98	2.45E-86	1.29E-28	1.3E-104	0.384532	6.49E-34	5.81E-38	1.634913	1.876108	4.352503
	std	1.31E-98	5.47E-87	3.16E-29	3.2E-105	0.059649	1.65E-34	1.4E-38	0.365578	0.493188	0.752218
	median	6.1E-100	1.3E-107	1.15E-29	1.9E-108	0.262132	8.24E-35	9.09E-39	5.12E-08	0.629908	3.420186
	rank	2	3	6	1	8	5	4	7	9	10
F3	mean	3E-160	3.9E-155	1.81E-11	22714.9	14.45275	9.03E-15	7.71E-25	458.1282	823.2796	2226.696
	best	4.5E-176	2.2E-211	2.87E-22	4817.622	6.657451	7.25E-21	3.29E-29	220.9651	104.5552	980.0687
	worst	5.9E-159	7.9E-154	1.96E-10	42590.93	29.56967	7.33E-14	1.39E-23	946.1665	5355.683	4000.966
	std	1.3E-159	1.8E-154	4.82E-11	11444.62	5.417527	2.14E-14	3.1E-24	170.163	1559.173	768.4028
	median	3E-169	1.1E-185	3.45E-14	20517.06	14.36599	1.65E-16	1.27E-26	421.0273	239.112	2336.894
	rank	1	2	5	10	6	4	3	7	8	9
F4	mean	8.21E-95	5.17E-81	0.018161	54.28253	0.515134	2.83E-14	1.92E-30	1.64627	5.985735	2.936851
	best	5.7E-101	1E-114	0.000199	0.471599	0.241859	3.65E-16	6.88E-32	1.44E-08	3.100586	1.679785
	worst	1.07E-93	1.01E-79	0.143356	90.27805	1.072136	2.66E-13	1.16E-29	6.195437	10.7205	3.748705
	std	2.42E-94	2.25E-80	0.035739	29.17733	0.211895	6.34E-14	2.93E-30	1.816032	1.86324	0.557296
	median	3.57E-96	2.5E-98	0.00324	64.0342	0.45734	4.25E-15	9.53E-31	1.018224	5.918239	2.913385
	rank	1	2	5	10	6	4	3	7	9	8
F5	mean	1.95E-05	28.82975	28.22662	27.1502	501.54	26.51912	26.79187	30.01965	125.4782	492.2424
	best	7.06E-07	28.45044	26.28345	26.7543	27.042	26.053	25.96032	22.56211	24.03923	208.5465
	worst	0.000168	28.94833	28.90235	27.89692	2378.444	27.93585	28.76149	102.6722	393.7645	1604.035
	std	4.31E-05	0.119877	0.901644	0.324329	777.9588	0.542129	0.746435	17.15039	82.77179	305.156
	median	3.15E-06	28.85657	28.74324	27.05339	99.01213	26.21274	26.43624	26.25606	115.6179	417.9566
	rank	1	6	5	4	10	2	3	7	8	9
F6	mean	0	0	3.618884	0.082505	0.152899	0.639441	1.110825	1.09E-16	0.011722	35.96325
	best	0	0	2.54627	0.003367	0.075682	0.25026	0.576087	6.63E-17	8.3E-06	18.60975
	worst	0	0	4.55379	0.313055	0.258552	1.259716	1.676233	2.07E-16	0.081377	67.09581
	std	0	0	0.637042	0.111943	0.042619	0.287192	0.34823	3.93E-17	0.020578	13.41989
	median	0	0	3.679755	0.022878	0.152863	0.621787	1.109756	9.76E-17	0.002319	32.44921
	rank	1	1	8	4	5	6	7	2	3	9
F7	mean	5.65E-05	0.000367	0.006374	0.001232	0.012817	0.000827	0.001209	0.172205	0.161375	0.009612
	best	1.19E-05	6.11E-05	0.002481	4.06E-05	0.005631	0.000143	0.000242	0.038115	0.063906	0.005539
	worst	0.000141	0.000896	0.012474	0.005569	0.023129	0.00154	0.002635	2.106994	0.316143	0.015126
	std	3.79E-05	0.000259	0.002758	0.00153	0.003658	0.000417	0.000553	0.456622	0.07542	0.003363
	median	5.03E-05	0.000344	0.00633	0.000611	0.012639	0.00075	0.001131	0.056289	0.145264	0.009104
	rank	1	2	6	5	8	3	4	10	9	7

(F1, F3–F7) and the second rank for F2. These results clearly demonstrate the robustness and precision of HCO in exploiting the search space and converging efficiently to the global optimum.

Specifically, for F1, HCO reached a mean fitness of *2.3E−193*, outperforming all competitor algorithms by several orders of magnitude. Similarly, in F3 and F4, HCO produced near-zero objective values, confirming its capability for highly accurate convergence.

For functions F5–F7, the algorithm maintained both stability (very low standard deviations) and superior accuracy compared to other approaches such as AOA, WOA, and GWO.

The minor difference observed in F2 (ranked second) can be attributed to the high sensitivity of this function to population diversity. Still HCO achieved results close to the global best. Overall, the outcomes in this group confirm the dominant exploitation power and convergence reliability of the proposed algorithm.

3.3 Performance analysis on high-dimensional multimodal functions (F8–F13)

The high-dimensional multimodal benchmarks are used to examine an algorithm's ability to escape

Table 2. Optimization results of high-dimensional multimodal test functions

F		HCO	AOA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F8	mean	-8987.62	-4074.12	-6273.04	-10729.5	-8089.26	-6420.16	-5370.64	-2479.96	-6438.96	-8307
	best	-12569.5	-4984.71	-7221.17	-12569.5	-10021	-7702.67	-6605.55	-4026.57	-7591.84	-9149.32
	worst	-6230.99	-2808.69	-5345.28	-6550.37	-6782.76	-4985.46	-4618.25	-1820.57	-5243.78	-6252.73
	std	2037.073	592.8844	478.7214	1990.014	739.2718	675.4963	460.6597	573.6972	662.6975	786.7745
	median	-8295.75	-4115.06	-6281.45	-11761.5	-8174.61	-6464.92	-5324.52	-2384.07	-6300.43	-8565.94
	rank	2	9	7	1	4	6	8	10	5	3
F9	mean	0	16.52442	185.9837	0	103.5478	0.659378	0	36.16669	57.27313	54.23415
	best	0	0	121.3601	0	55.80118	0	0	19.89918	26.87008	31.41892
	worst	0	155.6049	281.3677	0	167.2203	7.453729	0	75.61637	77.61344	92.97468
	std	0	42.47672	47.66369	0	25.42632	2.04861	0	12.3945	13.95144	16.46886
	median	0	0	184.8297	0	93.09722	2.84E-14	0	33.33109	59.20053	55.09856
	rank	1	3	8	1	7	2	1	4	6	5
F10	mean	4.44E-16	11.9794	1.807283	3.46E-15	0.733189	1.47E-14	3.82E-15	8.59E-09	2.883269	3.581202
	best	4.44E-16	4.44E-16	1.47E-14	4.44E-16	0.096344	7.55E-15	4.44E-16	5.92E-09	0.931466	2.912184
	worst	4.44E-16	19.96677	3.327181	7.55E-15	1.807437	2.18E-14	4E-15	1.19E-08	4.383258	4.492819
	std	0	10.03524	1.531198	2.38E-15	0.648825	2.58E-15	7.94E-16	1.84E-09	0.796413	0.425123
	median	4.44E-16	19.96368	2.708394	4E-15	0.649548	1.47E-14	4E-15	8.4E-09	2.776296	3.675363
	rank	1	10	7	2	6	4	3	5	8	9
F11	mean	0	0	0.005399	0.007239	0.375098	0.000452	0	8.707739	0.144124	1.476443
	best	0	0	0	0	0.263512	0	0	3.410419	0.000148	1.163917
	worst	0	0	0.017577	0.144788	0.480075	0.009045	0	17.8877	0.450134	1.863132
	std	0	0	0.006468	0.032376	0.066284	0.002022	0	3.753862	0.136767	0.157269
	median	0	0	0	0	0.371966	0	0	8.067823	0.088192	1.446364
	rank	1	1	3	4	6	2	1	8	5	7
F12	mean	2.05E-11	0.860852	7.368119	0.006772	1.299535	0.047129	0.083373	0.243101	1.23886	0.14402
	best	1.23E-11	0.568296	0.781769	0.001211	0.000718	0.006531	0.048972	5E-19	0.000422	0.051001
	worst	3.04E-11	1.139022	18.8273	0.022714	4.137494	0.17502	0.175572	1.444208	5.066644	0.297823
	std	5.63E-12	0.16393	4.834945	0.005936	1.304163	0.037699	0.033792	0.441211	1.311396	0.073129
	median	1.93E-11	0.863735	8.070525	0.004331	1.100269	0.036949	0.07348	0.037113	0.933497	0.131321
	rank	1	7	10	2	9	3	4	6	8	5
F13	mean	3.3E-10	2.899549	3.099128	0.304235	0.067713	0.487805	1.026526	0.090303	2.607317	2.915366
	best	1.8E-10	2.650531	1.825455	0.006121	0.009805	0.098376	0.586935	4.14E-18	0.022052	1.073928
	worst	5.92E-10	2.991026	4.47095	0.832325	0.411465	0.883088	1.529907	1.597462	13.80881	8.489249
	std	1.01E-10	0.075338	0.754813	0.230862	0.095773	0.194439	0.243764	0.355868	3.494079	1.886298
	median	3.44E-10	2.906776	3.016877	0.298513	0.030027	0.472447	0.987341	1.11E-17	1.016281	2.335764
	rank	1	8	10	4	2	5	6	3	7	9

local minima and maintain diversity during the search process. They are characterized by a vast number of local optima, making them particularly challenging for many metaheuristic methods.

As shown in Table 2, the proposed HCO algorithm again exhibits exceptional performance, obtaining the first rank in five out of six functions in this category.

For functions such as F9, F10, F11, F12, and F13, HCO consistently achieved the best or near-best mean values with minimal variance, highlighting its capacity for global exploration and adaptive search.

For instance, in F9 and F11, HCO achieved a perfect mean of *0*, matching the theoretical global minimum—an indication of its powerful global search ability and avoidance of local entrapment. Similarly, in F10, HCO demonstrated nearly zero fitness values with negligible standard deviation, reflecting both precision and consistency across independent runs. In F12 and F13, the algorithm maintained the leading position despite the high dimensional complexity, proving its efficiency in managing large and nonlinear landscapes.

Compared to traditional methods such as PSO and GA, which showed significant variability and frequent convergence to local minima, HCO displayed superior balance between exploration and exploitation, and stronger adaptability to rugged landscapes.

3.4 Performance analysis on fixed-dimensional multimodal functions (F14–F23)

The fixed-dimensional multimodal functions test the algorithm's ability to handle non-separable, highly complex landscapes and to maintain robustness across diverse problem scales. These functions typically include numerous local optima within bounded search spaces, requiring the algorithm to efficiently explore and exploit in a controlled manner.

Table 3. Optimization results of fixed-dimensional multimodal test functions

F		HCO	AOA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F14	mean	0.998004	1.467941	9.913151	2.371539	0.998004	4.908091	0.998006	3.620407	3.508733	1.006472
	best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	1.009191	0.998004	0.998004
	worst	0.998004	5.928845	15.50382	10.76318	0.998004	12.67051	0.998037	7.874003	10.76318	1.074676
	std	2.04E-16	1.182548	4.263033	2.962676	4.13E-12	4.682548	7.44E-06	1.833608	2.902867	0.022279
	median	0.998004	1.002451	10.76318	0.998004	0.998004	2.982105	0.998004	3.073378	2.487068	0.998055
	rank	1	5	10	6	2	9	3	8	7	4
F15	mean	0.000313	0.000733	0.003686	0.000604	0.00459	0.002423	0.003366	0.006279	0.003455	0.007024
	best	0.000307	0.000383	0.000308	0.000309	0.000313	0.000307	0.000308	0.00169	0.000307	0.001128
	worst	0.000424	0.002053	0.020942	0.001491	0.020363	0.020363	0.020364	0.016151	0.056543	0.023543
	std	2.61E-05	0.000395	0.007287	0.000311	0.008095	0.006145	0.007329	0.004477	0.012506	0.006775
	median	0.000307	0.00061	0.000519	0.000537	0.000649	0.000308	0.000314	0.004239	0.000307	0.003662
	rank	1	3	7	2	8	4	5	9	6	10
F16	mean	-1.03163	-1.03162	-1.02847	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	best	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	worst	-1.03163	-1.03147	-1	-1.03163	-1.03163	-1.03163	-1.03162	-1.03163	-1.03163	-1.03162
	std	1.25E-16	3.43E-05	0.009735	6.16E-10	4.31E-08	3.51E-09	1.61E-06	1.14E-16	1.25E-16	2.84E-06
	median	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	rank	1	7	8	2	4	3	6	1	1	5
F17	mean	0.397887	0.405128	0.397905	0.397888	0.397887	0.397888	0.435584	0.397887	0.763064	0.440887
	best	0.397887	0.397887	0.397888	0.397887	0.397887	0.397887	0.397889	0.397887	0.397887	0.397887
	worst	0.397887	0.481642	0.397959	0.397889	0.397888	0.39789	1.03059	0.397887	2.791184	1.248083
	std	0	0.021514	2.03E-05	4.53E-07	5.43E-08	7E-07	0.142518	0	0.874547	0.189997
	median	0.397887	0.398105	0.397897	0.397888	0.397887	0.397888	0.398005	0.397887	0.397887	0.397895
	rank	1	6	5	3	2	4	7	1	9	8
F18	mean	3	3.541195	12.45003	3.000039	3.000001	3.000008	3.000005	3.005772	3	4.366564
	best	3	3	3.000001	3	3	3	3	3	3	3
	worst	3	8.042265	84.00041	3.000418	3.000002	3.000024	3.000034	3.115444	3	30.00001
	std	7.79E-15	1.309006	25.19923	0.0001	4.75E-07	6.79E-06	9.34E-06	0.025814	2.99E-15	6.033711
	median	3	3.000951	3.00001	3.000002	3	3.000007	3.000001	3	3	3.000174
	rank	2	8	10	6	3	5	4	7	1	9
F19	mean	-3.86278	-3.85618	-3.86232	-3.86022	-3.86278	-3.8623	-3.86207	-3.47151	-3.82413	-3.86271
	best	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	-3.86277	-3.79417	-3.86278	-3.86278
	worst	-3.86278	-3.76133	-3.85489	-3.85202	-3.86278	-3.8575	-3.85476	-2.84716	-3.08976	-3.86193
	std	2.36E-15	0.022504	0.00175	0.003309	1.8E-07	0.001458	0.001741	0.305365	0.172852	0.000194
	median	-3.86278	-3.86258	-3.86271	-3.86182	-3.86278	-3.86277	-3.86251	-3.59265	-3.86278	-3.86278
	rank	1	8	4	7	2	5	6	10	9	3
F20	mean	-3.322	-3.03236	-3.24596	-3.21715	-3.24457	-3.27308	-3.27985	-1.2924	-3.26522	-3.22126
	best	-3.322	-3.31342	-3.32144	-3.32192	-3.32199	-3.32199	-3.31789	-2.40279	-3.322	-3.31969
	worst	-3.322	-2.54097	-3.0886	-2.84042	-3.2019	-3.13732	-3.19327	-0.64676	-3.13764	-2.95038
	std	9.55E-15	0.205686	0.088223	0.1373	0.058288	0.070525	0.050678	0.500668	0.065936	0.10484
	median	-3.322	-3.06475	-3.31973	-3.31963	-3.20305	-3.32199	-3.30302	-1.25019	-3.322	-3.25629
	rank	1	9	5	8	6	3	2	10	4	7
F21	mean	-10.1532	-7.71912	-5.42616	-8.57457	-6.86556	-9.90001	-6.66638	-2.94736	-4.27721	-5.56884
	best	-10.1532	-10.1521	-10.1094	-10.153	-10.1532	-10.1532	-9.71864	-5.0552	-10.1532	-9.55993
	worst	-10.1532	-3.22144	-2.61354	-5.03229	-2.63047	-5.09987	-4.15447	-0.59776	-2.63047	-2.42726
	std	8.08E-14	2.430227	3.458543	2.372369	2.848213	1.129836	1.88236	2.16359	3.061064	2.723362
	median	-10.1532	-8.88108	-2.67266	-10.1453	-5.10076	-10.1527	-6.69424	-2.96859	-2.68286	-5.88169
	rank	1	4	8	3	5	2	6	10	9	7
F22	mean	-10.4029	-7.29766	-7.9356	-8.5903	-9.08025	-9.87509	-8.48687	-6.54555	-7.51084	-6.91224
	best	-10.4029	-10.397	-10.313	-10.4028	-10.4029	-10.4029	-10.1281	-10.4029	-10.4029	-10.1194
	worst	-10.4029	-3.36366	-2.73631	-2.76589	-5.08766	-5.12844	-6.59636	-5.08767	-1.83759	-2.44031
	std	5.35E-14	2.755641	3.248829	2.875535	2.350401	1.623291	0.983481	2.355452	3.682769	2.819008
	median	-10.4029	-7.57095	-10.1685	-10.4008	-10.4029	-10.4026	-8.63625	-5.08767	-10.4029	-7.66295
	rank	1	8	6	4	3	2	5	10	7	9
F23	mean	-10.5364	-7.97581	-7.30225	-8.28907	-7.9985	-10.536	-7.54139	-9.31977	-6.35092	-7.43416
	best	-10.5364	-10.5129	-10.5069	-10.5363	-10.5364	-10.5363	-10.481	-10.5364	-10.5364	-10.484
	worst	-10.5364	-4.31874	-1.67493	-1.6765	-2.80663	-10.5356	-4.54951	-2.42734	-1.67655	-2.42545
	std	4.49E-14	2.140709	3.865176	3.191153	2.923208	0.000191	1.756497	2.55353	3.939688	3.161997
	median	-10.5364	-7.73744	-10.1913	-10.5301	-10.5363	-10.536	-7.93369	-10.5364	-4.50554	-8.81204
	rank	1	6	9	4	5	2	7	3	10	8

Received: October 2, 2025. Revised: November 15, 2025.

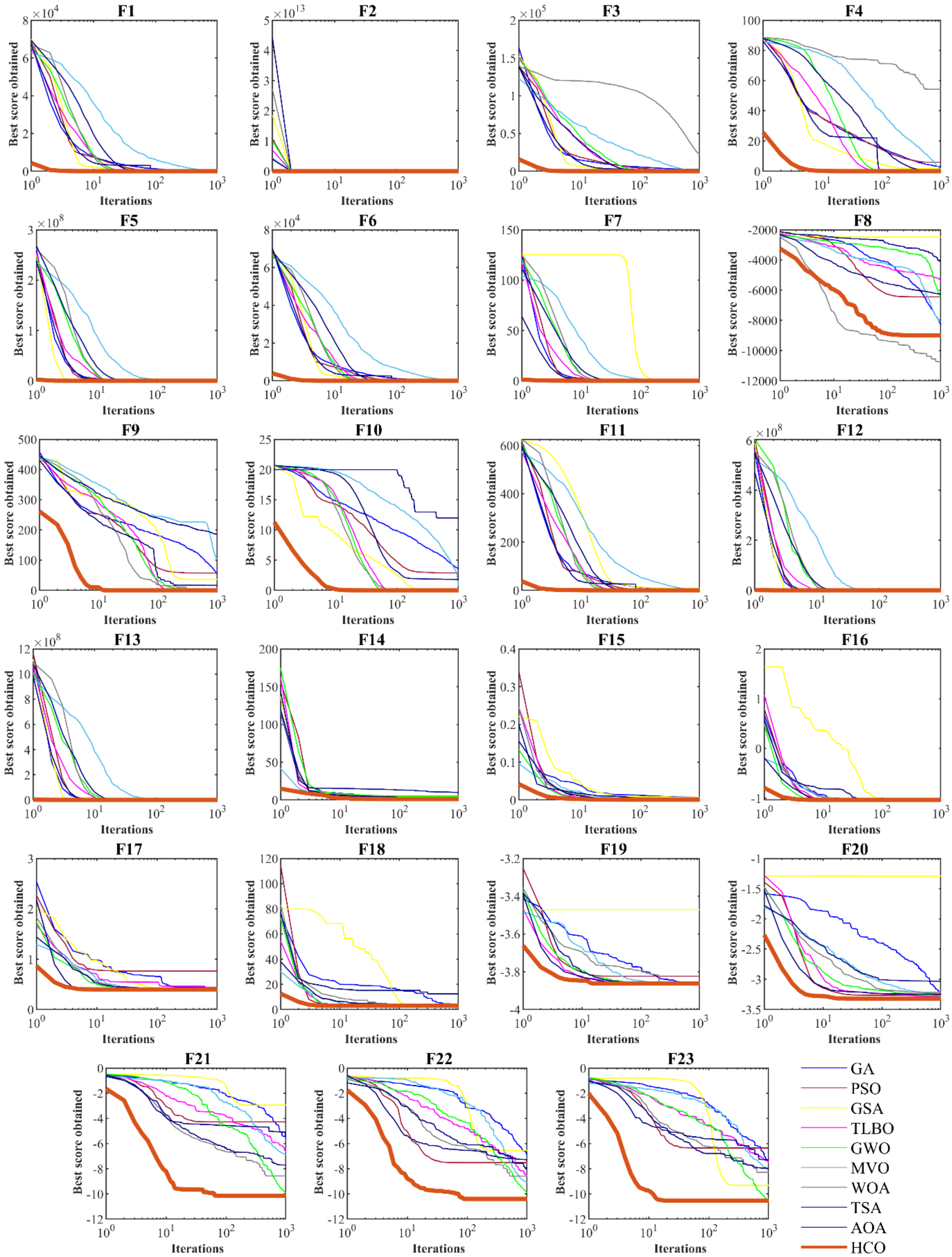

Figure. 1 Convergence curves obtained from HCO and competing algorithms on F1 to F23

Based on the results in Table 3, the proposed HCO algorithm achieved the first rank in nine out of ten test functions (F14–F23), clearly outperforming all comparative algorithms. Such dominance occuring across this benchmark category signifies that HCO possesses exceptional adaptability and equilibrium between diversification and intensification mechanisms.

In functions like F14, F16, F17, and F18, HCO reached values extremely close to the known global optima with almost zero standard deviation, highlighting its robust convergence and reproducibility across independent runs. The remaining algorithms (AOA, WOA, MVO, etc.) exhibited higher variability and slower convergence, indicating their relative instability in complex multimodal scenarios.

Notably, in F15, the mean value obtained by HCO (*3.13E−04*) outperformed all competitors, confirming its superior fine-tuning capability in narrow search basins. The performance consistency across all ten functions provides strong evidence that HCO is not only efficient but also highly reliable for a wide range of real-world optimization tasks.

3.5 Convergence analysis

Convergence trajectories of the proposed HCO algorithm and the nine comparative algorithms are illustrated in Fig. 1 for the benchmark functions F1–F23. These plots provide a comprehensive visualization of the dynamic search behaviour, convergence rate, and stability of each algorithm across both unimodal and multimodal landscapes.

As observed, HCO exhibits a consistently faster and smoother convergence pattern than its competitors in almost all test functions. In unimodal problems (F1–F7), HCO rapidly approaches the global optimum within the early iterations, demonstrating its high exploitation efficiency and capability for precise local refinement. The convergence curves of HCO in these cases show a sharp decline followed by an early stabilization near the global optimum, indicating strong convergence stability and the absence of oscillatory or premature stagnation behavior.

For high-dimensional multimodal functions (F8–F13), HCO maintains an effective exploration process during the initial stages, followed by a controlled acceleration toward optimal regions. Unlike several algorithms such as GSA and PSO that exhibit premature convergence or stagnation in local minima, HCO continues improving its best solutions through adaptive exploration. This trend confirms the algorithm's resilience against local entrapment and its ability to sustain diversified search dynamics.

In fixed-dimensional multimodal problems (F14–F23), the convergence patterns further reveal the balanced nature of HCO's search mechanism. The algorithm demonstrates steady progress with a well-regulated transition from exploration to exploitation, gradually refining the search space until convergence. Particularly in complex multimodal landscapes such as F21–F23, HCO achieves faster and more stable convergence trajectories than all other competitors, underscoring its adaptability to diverse objective function topologies.

Overall, the convergence profiles substantiate that HCO effectively balances exploration at early iterations with exploitation at later stages, leading to fast, stable, and high-precision convergence. This superior behavior results from the algorithm's biologically inspired mechanisms of cooperative movement, reproductive competition, and adaptive intensity control, which collectively enhance its optimization capability across different problem categories.

3.7 Discussion

The comprehensive analysis provided in Tables 1–3 reveals that the HCO algorithm surpasses all competing approaches in terms of accuracy, stability, and convergence behaviour. Its performance dominance—reflected by first-place ranks in 20 out of 23 test functions—demonstrates a remarkable balance between exploration and exploitation phases.

The algorithm's adaptive mechanism, inspired by the foraging and survival strategies of horseshoe crabs, enables it to dynamically adjust its movement and population interactions, preventing premature convergence and promoting effective information exchange among individuals.

The convergence curves illustrated in Fig. 1 further confirm that HCO consistently achieves faster and smoother convergence trajectories compared to all other methods. While algorithms such as WOA and AOA exhibit stagnation or oscillation in later iterations, HCO maintains a stable downward trend, reaching the global optimum efficiently.

In summary, the simulation results provide compelling evidence that the proposed HCO algorithm is a powerful, versatile, and robust metaheuristic optimization approach, capable of outperforming classical and modern competitors across a wide range of benchmark scenarios.

4. Conclusions and future work

In this study, the Horseshoe Crab Optimization algorithm, a novel population-based metaheuristic inspired by the reproductive and foraging behaviors of horseshoe crabs in tidal ecosystems is introduced. By mathematically modeling spawning aggregation and satellite male strategies, HCO effectively integrates global exploration with local exploitation, achieving a biologically motivated balance that drives efficient search dynamics. The algorithm's adaptive mechanisms—modulated by tidal-phase oscillations and stochastic perturbations—enable individuals to explore wide regions of the search space while refining solutions around promising candidates, mimicking the cooperative and competitive behaviors observed in nature.

Extensive simulation studies on 23 benchmark functions, spanning unimodal, high-dimensional multimodal, and fixed-dimensional multimodal landscapes, demonstrated that HCO consistently outperforms nine well-established metaheuristic algorithms, including AOA, WOA, MVO, GWO, TLBO, GSA, PSO, GA, and TSA. Statistical analysis revealed that HCO achieved first-place rank in 20 out of 23 functions, with extremely low variance and rapid convergence, highlighting its robustness, stability, and precision across diverse optimization challenges. The convergence analysis confirmed that HCO maintains a smooth, fast, and stable trajectory, effectively avoiding premature convergence and local entrapment even in highly rugged landscapes. These results substantiate the algorithm's versatility and reliability for complex, real-world optimization tasks.

Future research directions include the application of HCO to large-scale and multi-objective optimization problems, where the algorithm's adaptive exploration–exploitation dynamics could further demonstrate its efficacy. Incorporating hybrid strategies with complementary metaheuristics or integrating dynamic parameter adaptation schemes could enhance convergence speed and solution diversity. Additionally, extending HCO to constrained optimization domains, real-time adaptive systems, and uncertain environments may unlock its potential in engineering, logistics, and computational intelligence applications. Finally, theoretical analyses of convergence guarantees and parameter sensitivity would provide deeper insights into the underlying mechanisms that contribute to the algorithm's performance, facilitating its broader adoption and practical deployment.

In conclusion, the Horseshoe Crab Optimization algorithm represents a novel, biologically inspired, and highly effective optimization paradigm, capable of achieving superior performance across a wide spectrum of test scenarios, offering promising avenues for both theoretical and applied research in metaheuristic optimization.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, A.A, O.A.S, B.B, and Z.M; methodology, M.D, Z.M, A.A, G.B, and K.E; software, K.E and O.P.M; validation, K.E, M.D, O.P.M, and G.B; formal analysis, Z.M, M.D, K.E, B.B, and O.P.M; investigation, A.A, G.B, and Z.M; resources, O.A.S, Z.M, and B.B; data curation, Z.M and K.E; writing—original draft preparation, A.A, M.D, and O.A.S; writing—review and editing, Z.M, G.B, and K.E; visualization, K.E and O.P.M; supervision, M.D; project administration, K.E, O.A.S, O.P.M, B.B, and A.A; funding acquisition, K.E.

References

[1] J. L. J. Pereira, G. A. Oliver, M. B. Francisco, S. S. Cunha, and G. F. Gomes, "A Review of Multi-objective Optimization: Methods and Algorithms in Mechanical Engineering Problems", *Archives of Computational Methods in Engineering,* Vol. 29, No. 4, pp. 2285-2308, 2022, doi: 10.1007/s11831-021-09663-x.

[2] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences,* Vol. 179, No. 13, pp. 2232-2248, 2009, doi: 10.1016/j.ins.2009.03.004.

[3] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems", *Knowledge-Based Systems,* Vol. 96, pp. 120-133, 2016, doi: 10.1016/j.knosys.2015.12.022.

[4] R. Abu-Gdairi, R. Mareay, and M. Badr, "On Multi-Granulation Rough Sets with Its Applications", *Computers, Materials & Continua,* Vol. 79, No. 1, pp. 1025-1038, 2024.

[5] H. Qawaqneh, "New contraction embedded with simulation function and cyclic (α, β)-admissible in metric-like spaces", *International Journal of Mathematics and Computer Science,* Vol. 15, No. 4, pp. 1029-1044, 2020.

[6] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm", *Computers & Structures,* Vol. 169, pp. 1-12, 2016, doi: 10.1016/j.compstruc.2016.03.001.

[7] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications", *Future Generation Computer Systems,* Vol. 97, pp. 849-872, 2019, doi: 10.1016/j.future.2019.02.028.

[8] M. Braik, A. Sheta, and H. Al-Hiary, "A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm", *Neural Computing and Applications,* Vol. 33, No. 7, pp. 2515-2547, 2021, doi: 10.1007/s00521-020-05145-6.

[9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer", *Advances in engineering software,* Vol. 69, pp. 46-61, 2014.

[10] A. M. Zalzala, R. K. Jawad, A. A. M. M. AL-Salih, M. A. Ahmed, and I. K. Ibraheem, "Application of Spider-Tailed Horned Viper Optimization for Unit Commitment Problem in Power Systems", *INASS Express,* Vol. 1, No. 1, pp. 10-18, 2025, doi: 10.22266/inassexpress.2025.002.

[11] F. Werner, B. Batiha, T. Hamadneh, G. M. Gharib, W. Aribowo, and M. Dehghani, "Application of the Builder Optimization Algorithm for Sustainable Lot Size Optimization in Supply Chain Management: A Comprehensive Analysis and Comparison with Metaheuristic Approaches", *INASS Express,* Vol. 1, No. 1, pp. 39-50, 2025, doi: 10.22266/inassexpress.2025.005.

[12] Z. Monrazeri, O. P. Malik, and M. Dehghani, "Capacitor Placement in Power Distribution Networks for Voltage Profile Improvement and Loss Reduction Using Revolution Optimization Algorithm", *INASS Express,* Vol. 1, No. 1, pp. 29-38, 2025, doi: 10.22266/inassexpress.2025.004.

[13] T. Hamadneh, B. Batiha, G. M. Gharib, and W. Aribowo, "Application of Orangutan Optimization Algorithm for Feature Selection Problems", *INASS Express,* Vol. 1, No. 1, pp. 1-9, 2025, doi: 10.22266/inassexpress.2025.001.

[14] T. Hamadneh, B. Batiha, and G. M. Gharib, "Enhanced Distributed Generation Placement Using Carpet Weaver Optimization: A Comparative Metaheuristic Approach", *INASS Express,* Vol. 1, No. 1, pp. 19-28, 2025, doi: 10.22266/inassexpress.2025.003.

[15] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm", *Scientific Reports,* Vol. 14, No. 1, p. 5032, 2024, doi: 10.1038/s41598-024-54910-3.

[16] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization", *Engineering Applications of Artificial Intelligence,* Vol. 90, p. 103541, 2020, doi: 10.1016/j.engappai.2020.103541.

[17] Y. Xiao, H. Cui, R. A. Khurma, and P. A. Castillo, "Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems", *Artificial Intelligence Review,* Vol. 58, No. 3, p. 84, 2025, doi: 10.1007/s10462-024-11023-7.

[18] Z. Benmamoun, K. Khlie, G. Bektemyssova, M. Dehghani, and Y. Gherabi, "Bobcat Optimization Algorithm: an effective bio-inspired metaheuristic algorithm for solving supply chain optimization problems", *Scientific Reports,* Vol. 14, No. 1, p. 20099, 2024, doi: 10.1038/s41598-024-70497-1.

[19] N. A. Mansour, M. S. Saraya, and A. I. Saleh, "Groupers and moray eels (GME) optimization: a nature-inspired metaheuristic algorithm for solving complex engineering problems", *Neural Computing and Applications,* Vol. 37, No. 1, pp. 63-90, 2025, doi: 10.1007/s00521-024-10384-y.

[20] M. Azizi, U. Aickelin, H. A. Khorshidi, and M. Baghalzadeh Shishehgarkhaneh, "Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization", *Scientific Reports,* Vol. 13, No. 1, p. 226, 2023, doi: 10.1038/s41598-022-27344-y.

[21] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems", *Knowledge-Based Systems,* Vol. 259, p. 110011, 2023, doi: 10.1016/j.knosys.2022.110011.

[22] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning,* Vol. 3, No. 2, pp. 95-99, 1988, doi: 10.1023/A:1022602019183.

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

[24] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization", *IEEE Computational Intelligence Magazine,* Vol. 1, No. 4, pp. 28-39, 2006.

[25] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing", *Science,* Vol. 220, No. 4598, pp. 671-680, 1983.

[26] M. Dehghani *et al.*, "A Spring Search Algorithm Applied to Engineering Optimization Problems", *Applied Sciences,* Vol. 10, No. 18, p. 6173, 2020.

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

[28] T. Hamadneh *et al.*, "Barber Optimization Algorithm: A New Human-Based Approach for Solving Optimization Problems", *Computers, Materials & Continua,* Vol. 83, No. 2, 2025.

[29] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts game optimizer: A new optimization technique based on darts game", *International Journal of Intelligent Engineering and Systems,* Vol. 13, No. 5, pp. 286-294, 2020.

[30] Z. Montazeri, T. Niknam, J. Aghaei, O. P. Malik, M. Dehghani, and G. Dhiman, "Golf Optimization Algorithm: A New Game-Based Metaheuristic Algorithm and Its Application to Energy Commitment Problem Considering Resilience", *Biomimetics,* Vol. 8, No. 5, p. 386, 2023.

[31] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, "Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization", *Scientific Reports,* Vol. 13, No. 1, p. 10312, 2023, doi: 10.1038/s41598-023-37537-8.

[32] L. Wang, X. Wang, and Y. Pi, "Artificial Orca Optimiser: Theory and Applications for Global Optimisation Problems", *Expert Systems,* Vol. 42, No. 4, p. e70023, 2025, doi: 10.1111/exsy.70023.

[33] S. Chen, G. Yang, G. Cui, and X. Dong, "Raindrop optimizer: a novel nature-inspired metaheuristic algorithm for artificial intelligence and engineering optimization", *Scientific Reports,* Vol. 15, No. 1, p. 34211, 2025, doi: 10.1038/s41598-025-15832-w.

[34] V. Oberoi, M. Chankaya, M. Panda, and T. Chauhan, "Novel knowledge propagation optimization (KPO) algorithm for grid-tied PV system", *Results in Engineering,* Vol. 28, p. 107647, 2025, doi: 10.1016/j.rineng.2025.107647.

[35] Y. Peng *et al.*, "Wave Optics Optimizer: A novel meta-heuristic algorithm for engineering optimization", *Communications in Nonlinear Science and Numerical Simulation,* Vol. 152, p. 109337, 2026, doi: 10.1016/j.cnsns.2025.109337.

[36] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm", *Knowledge-Based Systems,* Vol. 191, p. 105190, 2020, doi: 10.1016/j.knosys.2019.105190.

[37] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm", *Expert Systems with Applications,* Vol. 149, p. 113338, 2020, doi: 10.1016/j.eswa.2020.113338.

[38] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications", *Advances in Engineering Software,* Vol. 114, pp. 48-70, 2017, doi: 10.1016/j.advengsoft.2017.05.014.

[39] Y. Jiang, Q. Wu, S. Zhu, and L. Zhang, "Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems", *Expert Systems with Applications,* Vol. 188, p. 116026, 2022, doi: 10.1016/j.eswa.2021.116026.

[40] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm", *Knowledge-Based Systems,* Vol. 89, pp. 228-249, 2015, doi: 10.1016/j.knosys.2015.07.006.

[41] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE transactions on evolutionary computation,* Vol. 1, No. 1, pp. 67-82, 1997.

[42] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster", *IEEE Transactions on Evolutionary computation,* Vol. 3, No. 2, pp. 82-102, 1999.

[43] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems", *Applied Intelligence,* Vol. 51, No. 3, pp. 1531-1551, 2021, doi: 10.1007/s10489-020-01893-z.

[44] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm", *Advances in Engineering Software,* Vol. 95, pp. 51-67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[45] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization", *Neural Computing and Applications,* Vol. 27, No. 2, pp. 495-513, 2016, doi: 10.1007/s00521-015-1870-7.