Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.052001

ARTICLE

BHJO: A Novel Hybrid Metaheuristic Algorithm Combining the Beluga Whale, Honey Badger, and Jellyfish Search Optimizers for Solving Engineering Design Problems

Farouq Zitouni[1,*], Saad Harous[2], Abdulaziz S. Almazyad[3], Ali Wagdy Mohamed[4,5], Guojiang Xiong[6], Fatima Zohra Khechiba[1] and KhadidjaÂ Kherchouche[1]

[1]Department of Computer Science and Information Technology, Kasdi Merbah University, P.O. Box 30000, Ouargla, Algeria

[2]Department of Computer Science, College of Computing and Informatics, University of Sharjah, P.O. Box 27272, Sharjah, United Arab Emirates

[3]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh, 11543, Saudi Arabia

[4]Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, 12613, Egypt

[5]Applied Science Research Center, Applied Science Private University, Amman, 11937, Jordan

[6]Guizhou Key Laboratory of Intelligent Technology in Power System, College of Electrical Engineering, Guizhou University, Guiyang, 550025, China

*Corresponding Author: Farouq Zitouni. Email: zitouni.farouq@univ-ouargla.dz

Received: 20 March 2024 Accepted: 09 July 2024 Published: 20 August 2024

ABSTRACT

Hybridizing metaheuristic algorithms involves synergistically combining different optimization techniques to effectively address complex and challenging optimization problems. This approach aims to leverage the strengths of multiple algorithms, enhancing solution quality, convergence speed, and robustness, thereby offering a more versatile and efficient means of solving intricate real-world optimization tasks. In this paper, we introduce a hybrid algorithm that amalgamates three distinct metaheuristics: the Beluga Whale Optimization (BWO), the Honey Badger Algorithm (HBA), and the Jellyfish Search (JS) optimizer. The proposed hybrid algorithm will be referred to as BHJO. Through this fusion, the BHJO algorithm aims to leverage the strengths of each optimizer. Before this hybridization, we thoroughly examined the exploration and exploitation capabilities of the BWO, HBA, and JS metaheuristics, as well as their ability to strike a balance between exploration and exploitation. This meticulous analysis allowed us to identify the pros and cons of each algorithm, enabling us to combine them in a novel hybrid approach that capitalizes on their respective strengths for enhanced optimization performance. In addition, the BHJO algorithm incorporates Opposition-Based Learning (OBL) to harness the advantages offered by this technique, leveraging its diverse exploration, accelerated convergence, and improved solution quality to enhance the overall performance and effectiveness of the hybrid algorithm. Moreover, the performance of the BHJO algorithm was evaluated across a range of both unconstrained and constrained optimization problems, providing a comprehensive assessment of its efficacy and applicability in diverse problem domains. Similarly, the BHJO algorithm was subjected to a comparative analysis with several renowned algorithms, where mean and standard deviation values were utilized as evaluation metrics. This rigorous comparison aimed to assess the performance of the BHJO algorithm about its counterparts, shedding light on its effectiveness and reliability in solving optimization problems. Finally, the obtained numerical statistics underwent rigorous analysis using the Friedman post hoc

Dunn's test. The resulting numerical values revealed the BHJO algorithm's competitiveness in tackling intricate optimization problems, affirming its capability to deliver favorable outcomes in challenging scenarios.

KEYWORDS

Global optimization; hybridization of metaheuristics; beluga whale optimization; honey badger algorithm; jellyfish search optimizer; chaotic maps; opposition-based learning

1 Introduction

Optimization is a fundamental problem-solving approach that aims to find the best possible solution from a set of feasible configurations [1]. It plays a crucial role in various domains, including engineering [2], economics [3], logistics [4], and data analysis [5]. The complexity of real-world optimization problems often arises from large search spaces, non-linear relationships, and multiple conflicting objectives [6].

To tackle such challenging optimization problems, researchers have developed a diverse range of techniques, including metaheuristic algorithms [7]. Metaheuristics are high-level problem-solving strategies that guide the search process by iteratively exploring and exploiting the search space to find optimal or near-optimal solutions [8]. Unlike exact optimization methods that guarantee optimal solutions but are limited to small-scale problems, metaheuristics are capable of handling large-scale and complex optimization problems [8].

Metaheuristic algorithms draw inspiration from natural phenomena, social behaviors, and physical processes to create intelligent search strategies [9]. These algorithms iteratively improve a population of candidate solutions by iteratively applying exploration and exploitation techniques. Exploration involves searching new regions of the search space to discover potential solutions, while exploitation focuses on refining and exploiting promising solutions to improve their quality [10]. A good balance between exploration and exploitation is essential for the success of metaheuristic algorithms in solving optimization problems [11]. Exploration allows the algorithm to search widely across the search space, uncovering diverse regions and potentially finding better solutions. It helps prevent the algorithm from getting stuck in local optimums and promotes global exploration. On the other hand, exploitation focuses on intensively exploiting the promising regions of the search space, refining and improving the solutions to converge towards optimal or near-optimal solutions. Balancing these two aspects ensures that the algorithm maintains a healthy exploration to discover new regions while exploiting the discovered promising solutions effectively. A well-balanced approach enables the algorithm to avoid premature convergence and thoroughly explore the search space to find high-quality solutions [9].

Metaheuristic algorithms can be broadly categorized into two groups based on their search strategy: population-based algorithms and individual-based algorithms [12]. Typically, population-based algorithms maintain a population of candidate solutions and explore the search space collectively, exchanging information between individuals to guide the search process. Examples of population-based algorithms include genetic algorithms [13] and particle swarm optimization [14]. On the other hand, individual-based algorithms, also known as trajectory-based algorithms, focus on improving a single solution or trajectory by iteratively modifying it through exploration and exploitation. Examples of individual-based algorithms include simulated annealing [15] and tabu search [16]. These algorithms often rely on a memory mechanism to keep track of previously visited regions and avoid getting trapped in local optimums.

Metaheuristic algorithms have demonstrated their effectiveness and versatility in solving a wide range of optimization problems in various fields. They have been successfully applied in areas such as engineering design [17], scheduling [18], resource allocation [19], data mining [20], and image processing and segmentation [21]. The ability of metaheuristic algorithms to handle complex, non-linear, and multi-objective optimization problems makes them particularly useful in real-world applications where traditional optimization techniques may fail to provide satisfactory results.

In this paper, we focus on the combination of three well-known metaheuristic algorithms, namely the beluga whale optimization [22], the honey badger algorithm [23], and the jellyfish search optimizer [24], to design a hybrid metaheuristic algorithm. We aim to leverage the strengths of these individual algorithms and propose a novel hybrid approach that improves optimization performance. The main contributions of this research work are:

1. After conducting an in-depth analysis of the three metaheuristics used to design the BHJO algorithm, we have concluded that the Beluga Whale Optimization (BWO) and the Honey Badger Algorithm (HBA) algorithms exhibit promising exploitation capabilities and stable exploration phases, although their balance between exploration and exploitation is suboptimal. Conversely, the Jellyfish Search (JS) algorithm demonstrates commendable exploration capacity and a well-balanced approach to exploration and exploitation, but its exploitation capability remains weak.
2. We incorporated Opposition-Based Learning into the BHJO algorithm to enhance its exploration capabilities and prevent it from getting trapped in local optimums.
3. We proposed a new balancing mechanism between exploration and exploitation, and it is very promising.

The organization of the remaining sections in the paper can be summarized in the following sentences. Section 2 reviews some recent metaheuristic algorithms and discusses their inspirations, highlighting the advancements in the field. Section 3 focuses on the working principle of the BHJO algorithm, providing a detailed explanation of its approach and highlighting its key features. Section 4 presents the experimental study conducted to evaluate the performance of the BHJO algorithm, including the methodology, benchmarks, and performance metrics employed. The results are analyzed and discussed in detail. Finally, Section 5 concludes the paper by summarizing the key findings, discussing their implications, and suggesting future directions for further research and improvements in hybrid metaheuristic algorithms.

2 Related Work

In this section, we provide an overview of several popular metaheuristic algorithms that have significantly contributed to the optimization field and have been published between 2020 and 2023. These algorithms have been extensively studied and applied in various domains to tackle complex optimization problems. The review focuses on highlighting the key features and principles of each algorithm. While the list is not exhaustive, it encompasses some of the most widely recognized and influential metaheuristics.

The Golden Eagle Optimizer (GEO) [25] is an optimization algorithm inspired by the hunting behavior and characteristics of golden eagles. The Mountain Gazelle Optimizer (MGO) [26] takes cues from the social life and hierarchy of wild mountain gazelles. The Artificial Ecosystem-based Optimization (AEO) [27] emulates the flow of energy in an ecosystem on the earth and mimics three unique behaviors of living organisms: production, consumption, and decomposition. The Dandelion Optimizer (DO) [28] reflects the process of dandelion seed long-distance flight relying on the wind: the rising stage, the descending stage, and the landing stage. The Archerfish Hunting Optimizer (AHO) [29] mimics the shooting and jumping behaviors of archerfish when hunting aerial insects. The African Vultures Optimization Algorithm (AVOA) [30] models after the African vultures' foraging and navigation behaviors. The Artificial Gorilla Troops Optimizer (GTO) [31] follows in the footsteps of the gorillas' collective lifestyle. The Beluga Whale Optimization [22] is inspired by the behaviors of pair swim, prey, and whale fall of Beluga whales. The Elephant Clan Optimization (ECO) [32] evokes the most essential individual and collective behaviors of elephants, such as powerful memories and high capabilities for learning. The Cheetah Optimizer (CO) [33] recreates the hunting strategies of cheetahs: searching, sitting and waiting, and attacking. The Bear Smell Search Algorithm (BSSA) [34] imitates both dynamic behaviors of bears: the sense of smell mechanism and the way bears move in search of food. The Gaining-Sharing Knowledge-based (GSK) [35] algorithm mimics the process of gaining and sharing knowledge during the human life span.

The Nutcracker Optimization Algorithm (NOA) [36] draws inspiration from two distinct behaviors of Clark's nutcrackers: search for seeds and storage in appropriate caches, and search for the hidden caches marked at different angles using markers. The Artificial Lizard Search Optimization (ALSO) [37] adopts the dynamic foraging behavior of Agama lizards and their effective way of capturing prey. The Red Deer Algorithm (RDA) [38] echoes the mating behavior of Scottish red deer in breeding seasons. The Ebola Optimization Search Algorithm (EOSA) [39] replicates the propagation mechanism of the Ebola virus disease. The Solar System Algorithm (SSA) [40] recreates the orbiting behaviors of some objects found in the solar system: i.e., the sun, planets, moons, stars, and black holes. The Siberian Tiger Optimization (STO) [41] imitates the natural behavior of the Siberian tiger during hunting and fighting. The Coati Optimization Algorithm (COA) [42] models the two natural behaviors of coatis: attacking and hunting iguanas and escaping from predators. The Artificial Rabbits Optimization (ARO) [43] simulates the survival strategies of rabbits in nature, including detour foraging and random hiding. The Red Piranha Optimization (RPO) [44] was inspired by the cooperation and organized teamwork of the piranha fish when hunting or saving their eggs. The Giza Pyramids Construction (GPC) [45] is based on the movements of the workers and pushing the stone blocks on the ramp when building the pyramids. The Sea-Horse Optimizer (SHO) [46] emulates the movement, predation, and breeding behaviors of sea horses in nature. The Aphid-Ant Mutualism (AAM) [47] mirrors the unique relationship between aphids and ants' species which is called mutualism.

The White Shark Optimizer (WSO) [48] mimics the behaviors of great white sharks, including their senses of hearing and smell, while navigating and foraging. The Circulatory System Based Optimization (CSBO) [49] simulates the function of the body's blood vessels with two distinctive circuits: pulmonary and systemic circuits. The Reptile Search Algorithm (RSA) [50] replicates the hunting behavior of Crocodiles: encircling, which is performed by high walking, and hunting, which is performed by hunting coordination. The Blood Coagulation Algorithm (BCA) [51] draws inspiration from the cooperative behavior of thrombocytes and their intelligent strategy of clot formation. The Color Harmony Algorithm (CHA) [52] mirrors the search behavior by combining harmonic colors based on their relative positions around the hue circle in the Munsell color system and harmonic templates. The Leopard Seal Optimization (LSO) [53] adopts the hunting strategy of the leopard seals. The Osprey Optimization Algorithm (OOA) [54] echoes the behavior of osprey in nature when hunting fish from the seas after detecting its position, then carries it to a suitable position to eat it. The Green Anaconda Optimization (GAO) [55] is inspired by the mechanism of recognizing the position of females by males during mating seasons and hunting strategies of green anacondas. The Subtraction-Average-Based Optimizer (SABO) [56] takes inspiration from the use of the subtraction average of searcher agents to update the position of population members in the search space. The Honey Badger Algorithm [23] simulates the behaviour of a honey badger when digging and finding honey. The Jellyfish Search [24] optimizer is motivated by the behavior of jellyfish in the ocean: following the ocean current, motions inside a jellyfish swarm, and convergence into jellyfish bloom.

Table 1 provides a detailed analysis of the metaheuristic algorithms discussed previously. It evaluates these algorithms by counting the instances of three fundamental steps that typically constitute metaheuristics: exploration, exploitation, and local minimum avoidance. This will allow readers to identify potential weaknesses in a specific algorithm. For instance, algorithms lacking the local minimum avoidance step can be located, suggesting a potential area for enhancement by incorporating this crucial phase. The first to third columns of Table 1 present information about the algorithm, and the fourth to sixth columns present the number of exploration, exploitation, and local minimum avoidance phases within each algorithm.

Table 1: Comparative analysis of metaheuristic algorithms based on exploration, exploitation, and local minimum avoidance

No.	Name	Abbreviation	Exploration	Exploitation	Local minimum avoidance
1	The artificial ecosystem-based optimization [27]	AEO	4	2	0
2	The bear smell search [34]	BSSA	2	2	0
3	The gaining-sharing knowledge algorithm [35]	GSK	1	1	0
4	The red deer algorithm [38]	RDA	2	2	1
5	The solar system algorithm [40]	SSA	2	4	1
6	The color harmony algorithm [52]	CHA	2	2	0
7	The golden eagle optimizer [25]	GEO	1	1	1
8	The African vultures optimization algorithm [30]	AVOA	2	4	0
9	The artificial lizard search optimization [37]	ALSO	2	2	0
10	The Giza pyramids construction [45]	GPC	1	1	0
11	The blood coagulation algorithm [51]	BCA	1	1	0
12	The jellyfish search [24]	JS	2	2	1
13	The dandelion optimizer [28]	DO	2	2	1

(Continued)

Table 1 (continued)

No.	Name	Abbreviation	Exploration	Exploitation	Local minimum avoidance
14	The Beluga whale optimization [22]	BWO	1	1	1
15	The cheetah optimizer [33]	CO	1	1	1
16	The Ebola optimization search algorithm [39]	EOSA	1	1	1
17	The siberian tiger optimization [41]	STO	2	2	0
18	The artificial rabbits optimization [43]	ARO	1	1	0
19	The aphid-ant mutualism [47]	AAM	3	3	1
20	The white shark optimizer [48]	WSO	2	1	0
21	The circulatory system based optimization [49]	CSBO	1	1	1
22	The reptile search algorithm [50]	RSA	2	2	0
23	The honey badger algorithm [23]	HBA	2	2	0
24	The nutcracker optimization algorithm [36]	NOA	2	2	1
25	The coati optimization algorithm [42]	COA	2	1	0
26	The red piranha optimization [44]	RPO	1	2	0
27	The sea-horse optimizer [46]	SHO	2	3	1
28	The leopard seal optimization [53]	LSO	1	2	0
29	The osprey optimization algorithm [54]	OOA	1	1	0
30	The green anaconda optimization [55]	GAO	1	1	1
31	The subtraction-average-based optimizer [56]	SABO	1	1	0

It is important to note that designing a new metaheuristic algorithm capable of effectively balancing the exploration of the search space with the exploitation of promising solutions, while also accurately determining the optimal times for each, is an extremely challenging task. Most hybridization efforts in the literature are geared towards addressing this issue by creating new hybrid algorithms that merge complementary strategies–for instance, combining a method with strong exploration but weak exploitation with one that excels in exploitation but not in exploration [57]. Additionally, the temporal complexity of these hybrid algorithms must be carefully considered. Hastily or poorly constructed hybrid models can significantly impair the efficiency of hybrid optimizers, underlining the necessity for meticulous development and evaluation.

It is worth emphasizing that the No-Free-Lunch (NFL) theorem [58] in optimization states that when considering the performance of optimization algorithms across all possible problem instances, no single algorithm can outperform all others on average. In other words, no universally superior algorithm can solve every optimization problem efficiently. The NFL theorem has implications for the field of metaheuristics and the existence of a large number of proposed algorithms. Since no single algorithm can be universally superior, researchers propose different metaheuristic algorithms to address specific problem characteristics and structures.

By developing a diverse set of metaheuristics, researchers aim to create algorithms that are tailored to specific problem domains and can exploit problem-specific information effectively. Different algorithms may have strengths and weaknesses that make them more suitable for particular problem types or classes. Therefore, the existence of a large number of metaheuristics can be seen as a way to cover a broad range of problem instances and improve the chances of finding a suitable algorithm for a given problem. Furthermore, the NFL theorem emphasizes the importance of algorithm selection and design for optimization problems. It suggests that the performance of an algorithm is intricately tied to the problem it is applied to. Therefore, the development of new metaheuristics aims to explore different search strategies, operators, or mechanisms that may be better suited for certain problem types or offer improvements over existing algorithms.

In summary, the No-Free-Lunch theorem implies that there is no universally superior optimization algorithm. This motivates the development of a large number of metaheuristic algorithms to address specific problem characteristics and improve the chances of finding suitable algorithms for various problem instances. The diversity of metaheuristics reflects the understanding that different algorithms may excel in different problem domains, and the search for the best algorithm for a specific problem requires careful consideration and experimentation.

3 Proposed Hybrid Algorithm

In this section, we provide a comprehensive explanation of the hybrid algorithm that we have proposed. Section 3.1 presents the source of inspiration that guided the development of our algorithm. Section 3.2 outlines the mathematical model underlying the BHJO algorithm. Finally, Section 3.3 presents the pseudo-code implementation of our algorithm along with a discussion of its time complexity.

3.1 Source of Inspiration

To elucidate the working principle of the proposed hybrid algorithm, a comprehensive understanding of the algorithms utilized in its design is provided. Section 3.1.1 focuses on elucidating the fundamental concept and workings of the BWO algorithm. Similarly, Section 3.1.2 delves into the underlying principles and mechanisms of the HBA algorithm. Finally, Section 3.1.3 provides a detailed account of the underlying idea behind the JS algorithm. By addressing the working principles of these constituent algorithms, a holistic understanding of the hybrid algorithm's operation is attained, thereby facilitating its effective application and potential future enhancements.

3.1.1 Beluga Whale Optimization

Beluga whales are marine mammals that belong to the whale family. They inhabit the Arctic and subarctic oceans and are characterized by their medium size. Beluga whales are known for their sociability, keen sensory perception, and distinct behaviors, including swimming with raised pectoral fins, synchronized diving and surfacing, and the release of bubbles from their blowholes. They have an omnivorous diet, consuming various prey such as fish and invertebrates, and often cooperate in groups for hunting and feeding. However, they face threats from natural predators like orcas and polar bears, as well as human activities. Additionally, when beluga whales die, their bodies sink to the ocean floor (known as a whale fall), creating a food source for deep-sea organisms[1]. The mathematical formulation of the exploration, the exploitation, and the concept of whale fall (i.e., local optimum avoidance) with the BWO algorithm is depicted in Fig. 1 and will be further elucidated in the following sections.

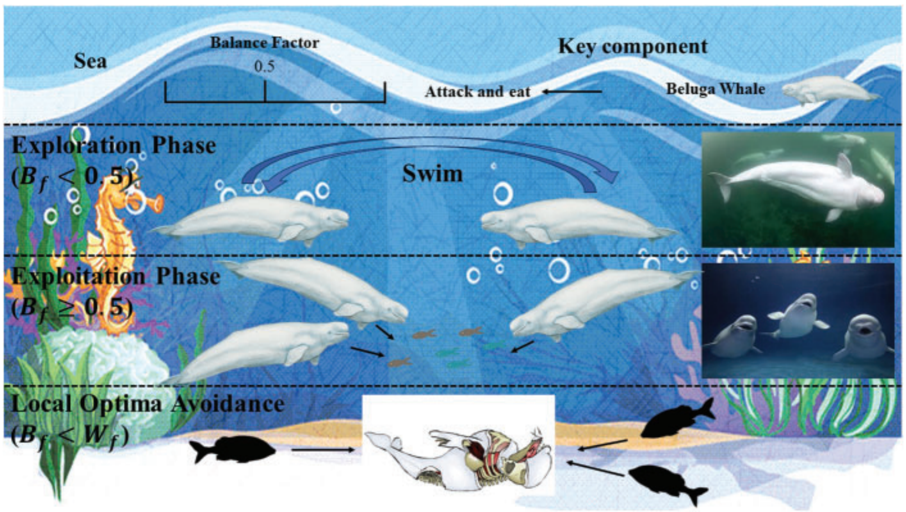

Figure 1: Behaviour of beluga whales [22]

The various symbols utilized in the equations about the BWO algorithm can be elucidated as follows:

- N: The population size.
- D: The dimensionality of the search space.
- LB: The lower bound of the search space.
- UB: The upper bound of the search space.
- $T_{\max}$: The maximum number of iterations.
- t: The current iteration.
- $X_{i,j}^t$: The position of the ith agent on the jth dimension at the iteration t.
- X_i^t: The position of the ith agent at the iteration t.
- ρ: A random number drawn from the range $\{1, \ldots, D\}$.
- r: A random number drawn from the range $\{1, \ldots, N\}$.
- r_1, r_2, r_3, r_4, r_5, r_6, r_7, α, and B_0: Random numbers drawn from the uniform distribution.
- X_{best}^t: The best solution in the current population.
- υ and ν: Random numbers drawn from the Gaussian distribution.
- β: A constant number set to 1.5.

[1]Encyclopedia of Life: https://eol.org/, accessed on 05 April 2023.

- Γ: The Gamma function.
- L_F: The lÃ©vy flight.

1) Initialization Phase: Eq. (1) is employed to initialize the different agents within the initial population of the BWO algorithm.

$$X_{i,j}^{0} = \alpha\,(\mathrm{UB} - \mathrm{LB}) + \mathrm{LB}\ ,\ i \in \{1, \ldots, N\}\ ,\ j \in \{1, \ldots, D\} \tag{1}$$

2) Exploration Phase: The locations of search agents are determined by the paired swimming behavior, where two beluga whales swim together in a synchronized or mirrored manner. This approach allows search agents to explore the search space more efficiently and effectively, leading to the discovery of new and potentially better solutions. The positions of the different agents are updated using Eq. (2).

$$\begin{cases} X_{i,j}^{t+1} = X_{i,\rho}^{t} + \left(X_{r,1}^{t} - X_{i,\rho}^{t}\right)(1 + r_1)\sin(2\pi r_2), & j = 2k \\ X_{i,j}^{t+1} = X_{i,\rho}^{t} + \left(X_{r,1}^{t} - X_{i,\rho}^{t}\right)(1 + r_1)\cos(2\pi r_2), & j = 2k + 1 \end{cases} \tag{2}$$

3) Exploitation Phase: The search agents share information about their current positions and consider the best solution, as well as other nearby solutions, when updating their locations. This mechanism helps the agents efficiently move towards promising regions of the search space. The exploitation phase also includes the lÃ©vy flight [59]. This strategy helps enhance search agents' convergence towards the global solution of the search space. The positions of agents are updated using Eq. (3).

$$X_{i}^{t+1} = r_3 X_{\text{best}}^{t} - r_4 X_{i}^{t} + 2r_4\left(1 - \frac{t}{T_{\max}}\right) L_F\left(X_{r}^{t} - X_{i}^{t}\right) \tag{3}$$

$$L_F = 0.05\left(\frac{\upsilon \times \sigma}{|\nu|^{\frac{1}{\beta}}}\right)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi \times \beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}}$$

4) Balance between Exploration and Exploitation: During the search process, the balance between the exploration and exploitation phases is determined by a swapping factor, denoted as B_f, which is calculated using Eq. (4). Therefore, if the value of B_f is greater than or equal to a specific threshold (e.g., 0.5), the search space is explored using Eq. (2); otherwise, the search space is exploited using Eq. (3).

$$B_f = B_0\left(1 - \frac{t}{2T_{max}}\right) \tag{4}$$

5) Whale Fall: The whale fall phase represents the concept of beluga whales' dying and becoming a food source for other creatures. In the BWO algorithm, the whale fall phase serves as a random operator to introduce diversity and prevent the search process from getting trapped in local optimums. The mathematical model of this step is expressed through Eq. (5).

$$X_{i}^{t+1} = r_5 X_{i}^{t} - r_6 X_{r}^{t} + r_7 X_{\text{step}} \tag{5}$$

$$X_{\text{step}} = (\text{UB} - \text{LB}) \exp\left(-C_2 \frac{t}{T_{\max}}\right)$$
$$C_2 = 2 \times W_f \times N$$
$$W_f = 0.1 - 0.05 \frac{t}{T_{\max}} \tag{6}$$

3.1.2 Honey Badger Algorithm

Honey badgers are captivating mammals renowned for their fearless and tenacious nature. They inhabit semi-deserts and rainforests across Africa, Southwest Asia, and the Indian subcontinent. With their distinct black and white fluffy fur, honey badgers typically measure between 60 to 77 centimeters in length and weigh around 7 to 13 kilograms. Their survival primarily relies on two modes: the digging mode and the honey mode. In the digging mode, honey badgers utilize their keen sense of smell to locate prey and engage in digging to capture it. In the honey mode, they take advantage of birds to identify beehives and obtain honey[2]. The mathematical formulation and associated concepts about the HBA algorithm are elaborated upon in the following sections.

The different symbols utilized in the equations related to the HBA algorithm can be described as follows:

- N: The population size.
- D: The dimensionality of the search space.
- LB: The lower bound of the search space.
- UB: The upper bound of the search space.
- $T_{\max}$: The maximum number of iterations.
- t: The current iteration.
- r_1, r_2, r_3, r_4, r_5, r_6, and r_7: Random numbers drawn from the uniform distribution.
- X_{prey}: The global best position so far found.
- β: A constant number replicating the ability of a honey badger to get food.
- d_i: The distance between the ith honey badger and X_{prey}.
- F: A flag number that alters the search direction. It is used to avoid getting stuck in local optimums.
- $\acute{C}$: A constant number greater than or equal to 1.

1) Initialization Phase: Eq. (7) is employed to initialize the different agents within the first population of the HBA algorithm.

$$X_{i,j}^{0} = r_1 (\text{UB} - \text{LB}) + \text{LB}\ ,\ i \in \{1, \ldots, N\}\ ,\ j \in \{1, \ldots, D\} \tag{7}$$

2) Smell Intensity: The intensity in the HBA algorithm is associated with the concentration strength of the prey's smell and its distance from a honey badger. The symbol I_i represents the smell intensity of the prey in relation to the i th honey badger. This intensity is determined using the Inverse Square Law [60], as illustrated in Fig. 2, and is defined by Eq. (8).

[2]Encyclopedia of Life: https://eol.org/, accessed on 05 April 2023.

$$I_i = r_2 \frac{S}{4\pi d_i^2} \tag{8}$$

$$S = \left(X_i^t - X_{i+1}^t\right)^2$$

$$d_i = X_{\text{prey}} - X_i^t$$

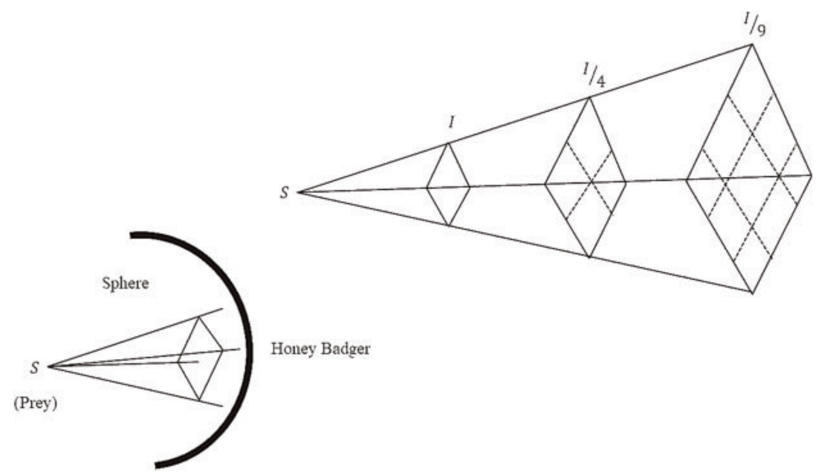

Figure 2: The inverse square law (I is the smell intensity, S is the prey's location, and r is a random number drawn from the uniform distribution) [23]

3) Density Factor: The density factor α is a randomizer that ensures the smooth transition from exploration to exploitation. Eq. (9) is used to compute α.

$$\alpha = \acute{C} \times \exp\left(\frac{-t}{T_{\max}}\right) \tag{9}$$

4) Exploration Phase: In nature, honey badgers dig like-cardioid shapes to surround and catch prey. Fig. 3 shows the form of a two-dimensional shape that has a heart-shaped curve. The exploration phase of the HBA algorithm is formulated using Eq. (10).

$$X_i^{t+1} = (1 + F\beta I_i) X_{\text{prey}} + F r_3 \alpha d_i \left|\cos(2\pi r_4)\left(1 - \cos(2\pi r_5)\right)\right| \tag{10}$$

$$F = \begin{cases} -1, & r_6 \le 0.5 \\ 1, & r_6 > 0.5 \end{cases}$$

5) Exploitation Phase: The exploitation phase is depicted by honey badgers' behaviors as they approach beehives, when following honey-guide birds. It is mathematically represented by Eq. (11).

$$X_i^{t+1} = X_{\text{prey}} + F r_7 \alpha d_i \tag{11}$$

3.1.3 Jellyfish Search Optimizer

Jellyfish, also known as medusae, are fascinating marine organisms that inhabit various depths and temperatures of water worldwide. They display an array of shapes, sizes, and colors, owing to their soft, translucent, and gelatinous composition. Typically, jellyfish possess a distinct bell-shaped body, contributing to their unique appearance. The size and form of the bell can significantly differ among jellyfish species. Additionally, jellyfish have developed a multitude of adaptation mechanisms to thrive in their oceanic environment. These mechanisms include specialized structures for hunting prey, such as elongated tentacles equipped with stinging cells known as nematocysts. Jellyfish employ two primary hunting strategies: passive swarming and active swarming. In the former, specific jellyfish follow the movement of a school of organisms, while in the latter, individuals hunt alone. During periods of non-hunting, jellyfish drift along with the ocean currents[3]. The behaviors of jellyfish are illustrated in Fig. 4. The mathematical formulation of exploration and exploitation concepts about the JS algorithm will be expounded upon in the following sections.

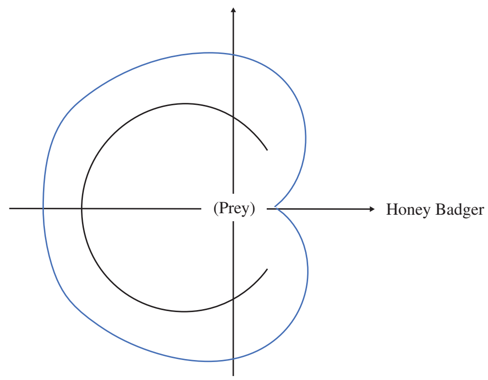

Figure 3: The digging behavior of honey badgers in nature to surround and catch prey [23]

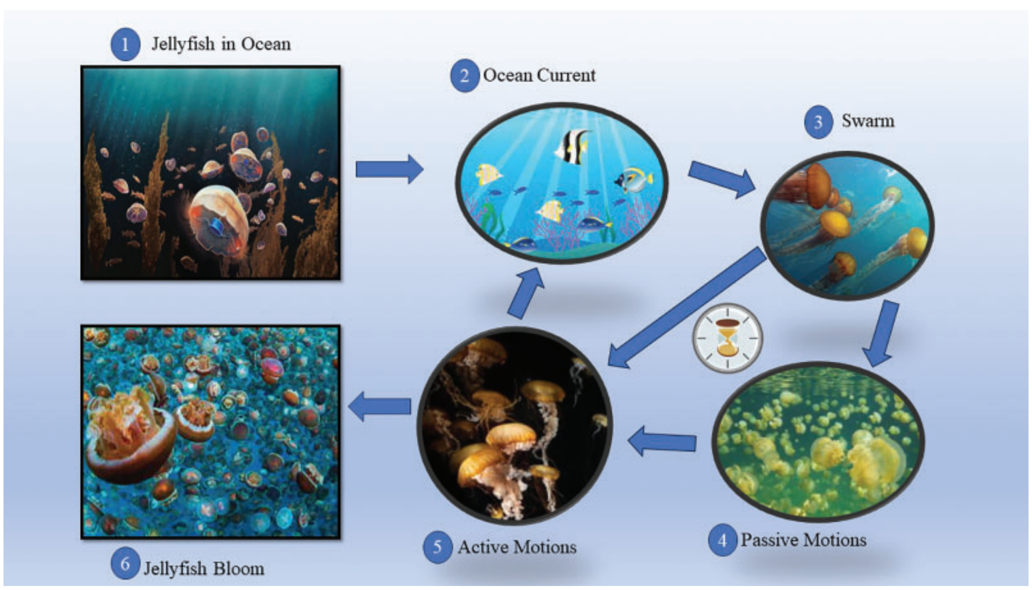

Figure 4: The behaviours of jellyfish in the ocean [24]

The symbols employed in the equations about the JS algorithm can be explained as follows:

- N: The population size.
- D: The dimensionality of the search space.

[3]Encyclopedia of Life: https://eol.org/, accessed on 05 April 2023.

- LB: The lower bound of the search space.
- UB: The upper bound of the search space.
- $T_{\max}$: The maximum number of iterations.
- t: The current iteration.
- X^t_{best}: The local best solution so far found.
- β: A number replicating the distribution coefficient.
- r_1, r_2, r_3, r_4, and r_5: Random numbers drawn from the uniform distribution.
- γ: A strictly positive number replicating the motion coefficient, and it is related to the length of motion around jellyfish's locations.

1) Initialization Phase: The initial population in the majority of metaheuristic algorithms is generally initialized randomly. However, this approach presents two significant drawbacks: a slow convergence and a tendency to get trapped in local optimums due to reduced population diversity. To address these issues and enhance diversity within the initial population, the authors of the JS algorithm conducted tests using various chaotic maps [61–63]. The findings revealed that the logistic map [64] produces a more diverse initial population compared to random initialization, thereby mitigating the problem of premature convergence [62,65]. Eq. (12) is utilized to initialize the different individuals within the initial population of the JS algorithm. It is worth pointing out that the values $\varphi_{1,\,j}$ are random numbers drawn from the uniform distribution and should satisfy the constraint: $\varphi_{1,j} \notin \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$.

$$X^0_{i,\,j} = \varphi_{i,\,j}\,(\text{UB} - \text{LB}) + \text{LB}\ ,\ i \in \{1, \ldots, N\}\ ,\ j \in \{1, \ldots, D\} \tag{12}$$

$$\varphi_{i+1,\,j} = 4\varphi_{i,\,j}\left(1 - \varphi_{i,\,j}\right)$$

2) Exploration Phase: Jellyfish utilize ocean currents to conserve energy and move efficiently. In the JS algorithm, this behavior is reflected in the exploration of the search space and the generation of new random candidate solutions. Consequently, the positions of individuals are updated using Eq. (13).

$$X^{t+1}_i = X^t_i + r_1\left(X^t_{\text{best}} - \beta r_2 M\right) \tag{13}$$

$$M = \frac{1}{N}\left(X^t_1 + \ldots + X^t_N\right)$$

3) Exploitation Phase: Jellyfish navigate within the swarm to search for food. In the JS algorithm, this corresponds to the exploitation of the search space. There are two distinct behaviors observed: a jellyfish either moves randomly in search of food independently or moves towards a better solution. These mechanisms are referred to as passive motion and active motion, respectively [66,67].

1. **Passive Motion:** The different agents in a specific population update their locations using Eq. (14).

   $$X^{t+1}_i = X^t_i + \gamma r_3(\text{UB} - \text{LB}) \tag{14}$$

2. **Active Motion:** To simulate this kind of movement, we randomly select two locations p and q. If the solution at the location p is better than q, then q moves towards p and vice versa. Hence, the different agents within a given population update their locations using Eq. (15). This exploitation mechanism has proven to be very efficient, according to the work reported

in [68].

$$X_i^{t+1} = X_i^t + r_4 \overrightarrow{\Delta} \tag{15}$$

$$\overrightarrow{\Delta} = \begin{cases} X_p^t - X_q^t, & f(X_p^t) < f(X_q^t) \\ X_q^t - X_p^t, & f(X_p^t) \geq f(X_q^t) \end{cases}$$

4) Time Control Mechanism: The JS algorithm incorporates a time control model that facilitates the transition between exploration and exploitation phases, as well as between passive and active motions. This time control mechanism is mathematically modeled by Eq. (16) and visually illustrated in Fig. 5.

$$c(t) = \left| (2r_5 - 1) \times \left(1 - \frac{t}{T_{\max}}\right) \right| \tag{16}$$

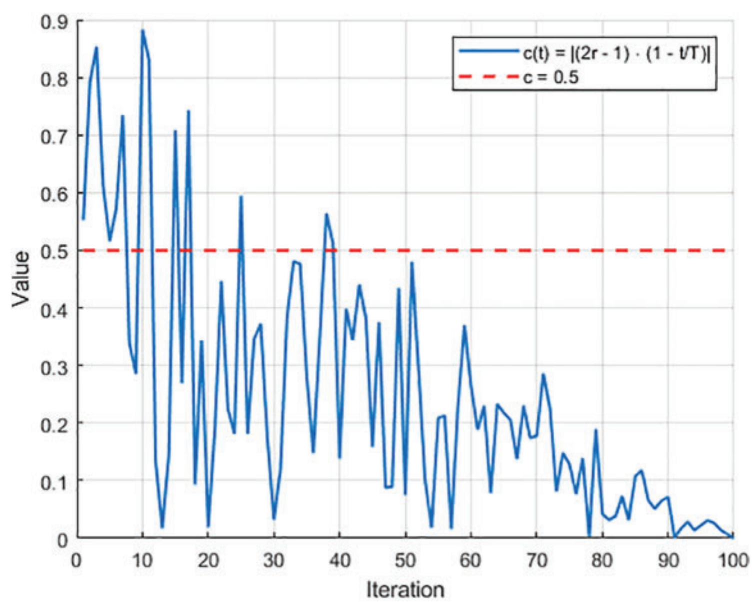

Figure 5: Time control mechanism used in the JS algorithm [24]

3.2 Mathematical Model of the BHJO Algorithm

By combining the previous algorithms, we aim to design a more robust and efficient hybrid meta-heuristic algorithm that can leverage the advantages and strengths of each one. Hybrid nature-inspired approaches involve combining different optimization algorithms to enhance their performance. There are two manners of hybridization, namely high-level and low-level [69]. In the first type, the algorithms to be hybridized are applied consecutively, whereas in the second type, the similar steps of each algorithm are merged.

It is worth emphasizing that the exploration and exploitation abilities of the aforementioned algorithms–namely, the Beluga Whale Optimization [22], the Honey Badger algorithm [23], and the Jellyfish Search Optimizer [24]–are quite good. However, the results reported in [22–24] indicate the following observations: (i) the exploitation of the BWO algorithm is better than its exploration; (ii) the exploitation of the HBA algorithm is better than its exploration; and iii) the exploration of the JS algorithm is better than its exploitation.

We have adopted a low-level hybridization approach to design the BHJO algorithm. Specifically, the exploration phases of the BWO, HBA, and JS algorithms are combined to form the exploration component of the BHJO optimizer. On the other hand, the exploitation phases of the BWO and HBA metaheuristics are utilized to design the exploitation component of the BHJO algorithm. The key to success often lies in finding the right combination that effectively balances the trade-off between exploration and exploitation. Exploration involves generating new solutions and discovering promising regions, while exploitation aims to improve existing solutions and avoid local optimums.

3.2.1 Initialization Phase

To improve the diversity of the population, the logistic map has been used to generate chaotic sequences [70]. This can enhance the global search ability of the BHJO algorithm by allowing it to explore the search space more effectively. Eq. (12) is employed to create the first population of our algorithm.

3.2.2 Exploration Phase

The exploration phase of the BHJO algorithm is composed of two strategies, namely the first exploration strategy and the second exploration strategy. In the first strategy, we combine the exploration of the HBA and the JS algorithms. In the second strategy, we use the exploration of the BWO algorithm. To switch between the first and second strategies, we employ the following technique: First, we assume a uniform random number; next, if this number is less than 0.5, we apply the first strategy; otherwise, we apply the second strategy.

1) First Exploration Strategy: We utilize Eq. (17) to perform the first exploration strategy. This equation is designed by merging Eqs. (10) and (13). The factor B_f is computed using Eq. (18).

$$X_i^{t+1} = r_1 X_{\text{prey}} + FI_i \overrightarrow{\Delta} d_i B_f \left|\cos(2\pi r_2)\left(1 - \cos(2\pi r_3)\right)\right| \tag{17}$$

2) Second Exploration Strategy: We employ Eq. (2) to execute the second exploration strategy.

3.2.3 Exploitation Phase

The exploitation phase of the BHJO algorithm works with two strategies, namely first exploitation strategy and the second exploitation strategy. In the first strategy, we employ the exploitation of the BWO algorithm. In the second strategy, we use the exploitation of the HBA algorithm. To switch between the first and second strategies, we use the following technique: First, we assume a uniform random number; next, if this number is greater than $1 - B_f$, we apply the first strategy; otherwise, we apply the second strategy. The factor B_f is computed using Eq. (18).

1) First Exploitation Strategy: We use Eq. (3) to carry out the first exploitation strategy.

2) Second Exploitation Strategy: We apply Eq. (11) to realize the second exploitation strategy.

3.2.4 Balance between Exploration and Exploitation

In most metaheuristic algorithms, the optimizers explore and then exploit the search space. This approach has several disadvantages. For instance, during the exploration phase, the algorithm searches a wide range of solutions to identify promising areas in the search space, but it may fail to find optimal or near-optimal solutions; or the exploitation phase, which focuses on intensifying the search in selected regions, can lead the algorithm to prematurely converge to a suboptimal solution, without exploring other regions of the search space that may contain better solutions. Therefore, we propose a novel search mechanism that allows for exploration across a wide range of solutions to identify promising areas in the research domain; and simultaneously, it intensifies research in those promising areas to ensure an effective balance between exploration and exploitation, ultimately leading to the discovery of optimal or near-optimal solutions. Fig. 6 depicts the curve of balancing between exploration and exploitation, which is defined by Eq. (18).

$$B_f = \left|(r_1 C - 1)\left(1 + m\cos\left(2\pi f \frac{t}{T_{max}}\right)\cos(2\pi f t)\right)\right| \tag{18}$$

where C, f, and m are constant numbers set to 2, 5, and −3, respectively; and r_1 is a uniform random number.

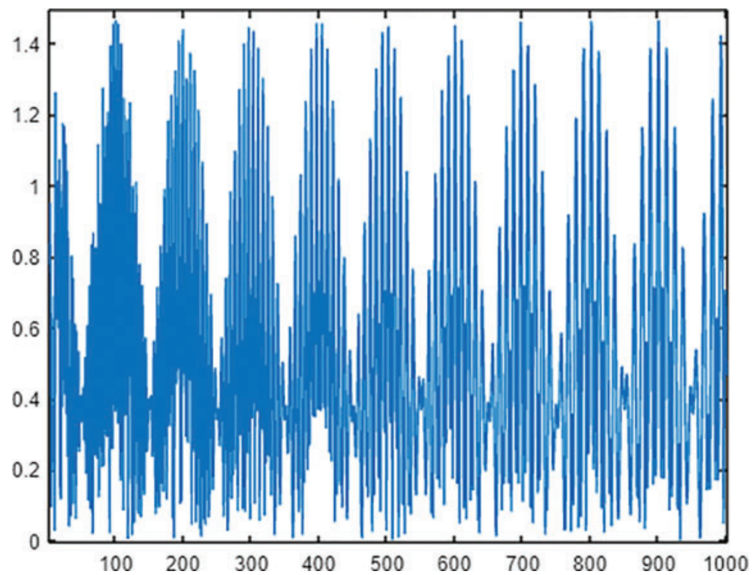

Figure 6: The curve of balancing between exploration and exploitation phases in the BHJO algorithm

3.2.5 Local Optimum Avoidance

To enhance the BHJO algorithm's capability to avoid getting stuck in local optimums, we utilize Eq. (5).

3.2.6 Opposition-Based Learning

The Combined Opposition-Based Learning (COBL) [71] is a novel research strategy that integrates two variants of opposition-based learning: Lens Opposition-Based Learning [72] and Random Opposition-Based Learning [73]. This research strategy is renowned for its ability to accelerate the convergence speed of metaheuristics by incorporating the concept of lens opposition, which entails generating an opposite solution based on the lens formed between the current solution and the global best solution. By integrating COBL into the BHJO algorithm, we can effectively mitigate the issues of local optimums and premature convergence. COBL is given by Eq. (19). It is worth emphasizing that

Eq. (19) is applied each time a candidate solution is created or updated.

$$\breve{X}_i = \begin{cases} \mathrm{LB} + \mathrm{UB} - r_1 X_i, & q < 0.5 \\ \dfrac{\mathrm{LB} + \mathrm{UB}}{2} + \dfrac{\mathrm{LB} + \mathrm{UB}}{2k} - \frac{X_i}{k}, & q \geq 0.5 \end{cases} \tag{19}$$

where q is a uniform random number and k is a scale factor set to 12000 according to [71].

3.3 Pseudo-Code and Time Complexity of the BHJO Algorithm

Algorithm 1 describes the different steps of the proposed BHJO algorithm. Theoretically, the computational complexity of the BHJO algorithm is an important metric to assess its performance. It includes three processes: fitness evaluation, initialization of the first population, and updating the positions of search agents. First, the time complexity of the fitness evaluation is $O(D) \sim O(n)$. Second, the time complexity of the initialization of the first population is $O(\max\{N,\ N \times D\}) \sim O(n^2)$. Finally, the time complexity of the swarming behavior is $O(T_{max} \times N \times D) \sim O(n^3)$. From the previous time complexities, we conclude that the time complexity of BHJO is $O(n^3)$. For further clarity, Fig. 7 provides the flowchart of the proposed algorithm.

Algorithm 1: The pseudo-code of the BHJO algorithm.

```
Input: Initialize the different parameters of the BHJO algorithm.
Input: Define the objective function to be minimized f (X).
/* Initialization of the first population */
Initialize the first population {X_1^0, ..., X_N^0} using Eq. (12);
for i ← 1 to N do
    Compute X̆_i^0, the opposite solution of X_i^0, using Eq. (19);
    X_i^0 ← argmin {f (X_i^0), f (X̆_i^0)};
    Check the boundaries of the solution X_i^0;
end
/* Swarming behavior of the BHJO algorithm */
for t ← 1 to T_max do
    Compute B_f using Eq. (18);
    for i ← 1 to N do
        /* Exploration of the search space */
        if (B_f ≥ 0.5) then
            /* First exploration strategy */
            if (rand(0, 1) < 0.5) then
                Compute the position X_i^t using Eq. (17);
            end
            /* Second exploration strategy */
            else
                Compute the position X_i^t using Eq. (2);
            end
        end
        /* Exploitation of the search space */
```

(Continued)

Algorithm 1 (continued)

```
        else
          /* First exploitation strategy                                   */
          if (rand(0, 1) > (1 − B_f)) then
              Compute the position X_i^t using Eq. (3);
          end
          /* Second exploitation strategy                                  */
          else
              Compute the position X_i^t using Eq. (11);
          end
        end
        Compute X̆_i^t, the opposite solution of X_i^t, using Eq. (19);
        Compute X̆_i^{t−1}, the opposite solution of X_i^{t−1}, using Eq. (19);
        X_i^t ← argmin {f (X_i^t), f (X̆_i^t), f (X_i^{t−1}), f (X̆_i^{t−1})};
        Check the boundaries of the solution X_i^t;
    end
    /*  Local optimums avoidance                                           */
    Compute W_f using Eq. (6);
    for i ← 1 to N do
        if B_f ≤ W_f then
            Compute the position X_i^t using Eq. (5);
            Compute X̆_i^t, the opposite solution of X_i^t, using Eq. (19);
            X_i^t ← argmin {f (X_i^t), f (X̆_i^t)};
            Check the boundaries of the solution X_i^t;
        end
    end
end
Output the best solution;
```

4 Experimental Results and Discussion

In this section, we delve into the experimental study, which aims to evaluate the performance of the BHJO algorithm on different fronts. Section 4.1 focuses on scrutinizing the BHJO algorithm's performance on benchmark test functions or Unconstrained Optimization Problems (UOPs), allowing us to assess its effectiveness and efficiency in solving standard optimization problems. Moving on to Section 4.2, our attention shifts to investigating the BHJO algorithm's performance on engineering design problems or Constrained Optimization Problems (COPs). This analysis enables us to gauge its applicability and robustness in tackling real-world optimization challenges specific to the field of engineering design. Through these comprehensive evaluations, we aim to gain insights into the BHJO algorithm's strengths, weaknesses, and potential areas of improvement.

The experiments were conducted using a computer system with the following hardware and software configuration. The computer is equipped with an Intel(R) Core(TM) i7-9750H CPU running at a base frequency of 2.60 GHz and a maximum turbo frequency of 4.50 GHz. The installed RAM capacity is 16 GB, providing ample memory for running complex algorithms and processing large computational loads. The operating system used is Windows 11 Home, which provides a stable and user-friendly environment for conducting the experiments. The experiments were implemented using the MATLAB R2020b programming language, known for its extensive mathematical and scientific computing capabilities. Furthermore, statistical analyses were conducted using IBM SPSS Statistics, a widely used software package for analyzing and interpreting data, facilitating robust statistical analysis.

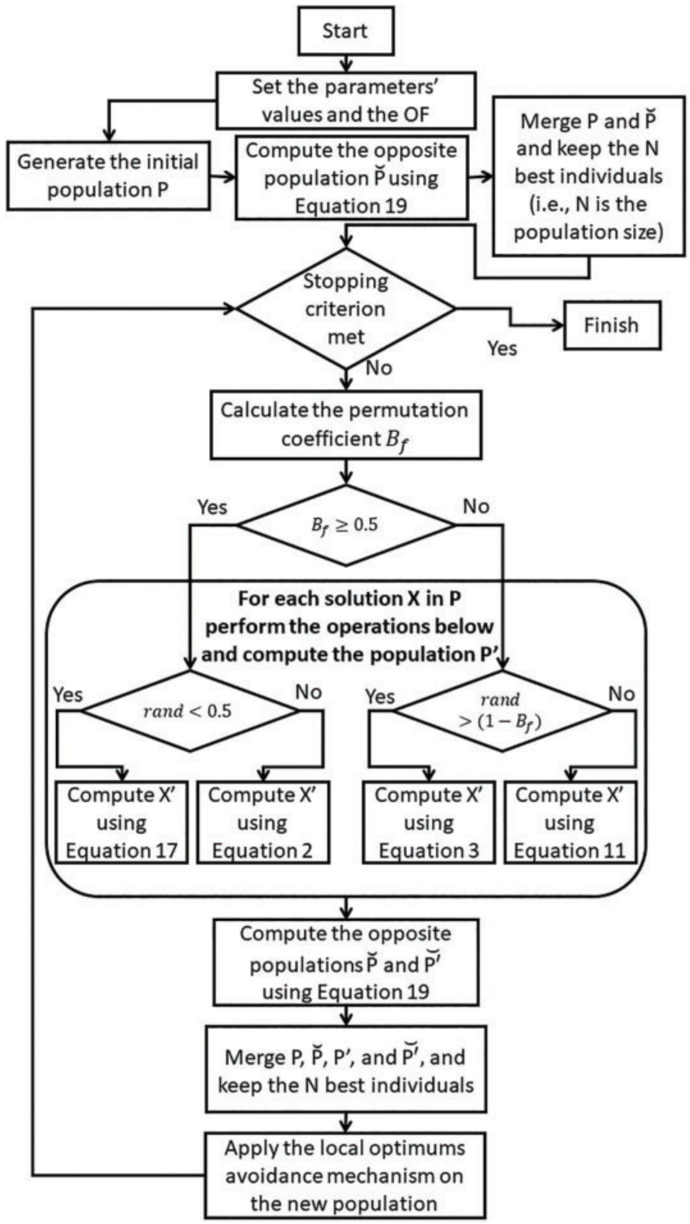

Figure 7: The flowchart of the proposed BHJO algorithm

4.1 Performance of the BHJO Algorithm on UOPs

To evaluate the effectiveness of the BHJO algorithm, a collection of 40 well-established benchmark test functions from the existing literature is selected [74,75]. The set covers four types of functions: unimodal, multimodal, hybrid, and composition functions. Tables 2–6 provide an overview of the details of each function, including their mathematical expressions, dimensions, ranges, and global optimums. Unimodal functions $\{F_1, \ldots, F_7\}$ have only one global optimum and are used to assess the exploitation ability of optimization methods. On the other hand, multimodal functions $\{F_8, \ldots, F_{20}\}$ possess several local optimums and are employed to evaluate the exploration ability of optimization methods, specifically the avoidance of local optimums and the prevention of premature convergence. Finally, the hybrid and composition functions $\{F_{21}, \ldots, F_{40}\}$ are known to be very challenging and difficult. They have a large number of local optimums and are used to evaluate the well-balanced trade-off between exploration and exploitation in metaheuristic algorithms.

Table 2: The description of unimodal functions

ID	Mathematical expression	Dimension	Range	Global optimum
F_1	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	{30, 50, 100, 1000}	[−100, 100]	0
F_2	$f(\mathbf{x}) = (\sum_{i=1}^{D} x_i^2)^2$	{30, 50, 100, 1000}	[−100, 100]	0
F_3	$f(\mathbf{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	{30, 50, 100, 1000}	[−100, 100]	0
F_4	$f(\mathbf{x}) = \max_i\{\lvert x_i \rvert, 1 < i < n\}$	{30, 50, 100, 1000}	[−100, 100]	0
F_5	$f(\mathbf{x}) = \sum_{i=1}^{D} i x_i^2$	{30, 50, 100, 1000}	[−100, 100]	0
F_6	$f(\mathbf{x}) = \sum_{j=1}^{D-4}[(x_{i-1} - 10x_i)^2$ $+5(x_{i+1} - x_{i+2})^2$ $+(x_i - xi + 1)^4$ $+10(x_{i-1} - x_{i+2})^4]$	{30, 50, 100, 1000}	[−4, 5]	0
F_7	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^{10}$	{30, 50, 100, 1000}	[−100, 100]	0

Table 3: The description of multimodal functions

ID	Mathematical expression	Dimension	Range	Global optimum
F_8	$f(\mathbf{x}) = 0.5 + \dfrac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	{30, 50, 100, 1000}	[−500, 500]	0
F_9	$f(\mathbf{x}) =$ $10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	{30, 50, 100, 1000}	[−5.12, 5.12]	0
F_{10}	$f(\mathbf{x}) =$ $-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})$ $-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 +$ $\exp(1)$	{30, 50, 100, 1000}	[−32, 32]	0

(Continued)

Table 3 (continued)

ID	Mathematical expression	Dimension	Range	Global optimum
F_{11}	$f(\mathbf{x}) = 1+\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod\cos(\frac{x_i}{\sqrt{i}})$	{30, 50, 100, 1000}	[−600, 600]	0
F_{12}	$f(\mathbf{x}) = x_1^2 + 10^6\sum_{i=2}^{D}x_i^2$	{30, 50, 100, 1000}	[−10, 10]	0
F_{13}	$f(\mathbf{x}) = 0.1(\sin(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2 [1 + \sin(3\pi x_{i+1})] + (x_n - 1)^2[1+ \sin(2\pi x_i)]) + \sum_{i=1}^{D} u(x_i,\ 5,\ 100,\ 4)$	{30, 50, 100, 1000}	[−50, 50]	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & if\ x_i > a \\ 0, & if - a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$				

Table 4: The description of fixed-multimodal functions

ID	Mathematical expression	Dimension	Range	Global optimum
F_{14}	$f(\mathbf{x}) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[−10, 10]	0
F_{15}	$f(\mathbf{x}) = -\sum_{j=1}^{m}(\sum_{i=1}^{4}(x_j - C_{ji})^2 + \beta_i)^{-1}$	4	[0, 10]	0
F_{16}	$f(\mathbf{x}) = -\frac{0.001}{[0.001^2 + (x_1 - 0.4x_2 - 0.1)^2]} - \frac{0.001}{[0.001^2 + (2x_1 + x_2 - 1.5)^2]}$	2	[−500, 500]	−2000
F_{17}	$f(\mathbf{x}) = (2x_1^3x_2 - x_2^3)^2 + (6x_1 - x_2^2 + x_2)$	2	[−500, 500]	0
F_{18}	$f(\mathbf{X}) = \sum_{i=1}^{11}[\alpha_i\frac{x_1(\beta_i^2 + \beta_i x_2)}{\beta_i^2}+\beta_i x_3+x_4]^2$	4	[−5, 5]	3
F_{19}	$f(\mathbf{x}) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 +90(x_4 - x_3^2)^2 + (1 - x_3)^2 +10.1(x_2 - 1)^2 + (x_4 - 1)^2 +19.8(x_2 - 1)(x_4 - 1)$	4	[−10, 10]	0
F_{20}	$f(\mathbf{x}) = 0.5x_1^2 + 0.5[1 - \cos 2x_1] + x_2^2$	2	[−500, 500]	0

$m = 10$

$\beta = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)^T$

$$C = \begin{pmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{pmatrix}$$

Table 5: The description of hybrid functions

ID	ID in CEC	Dimension	Range	Global optimum
F_{21}	Hybrid function F_{17} (N = 4) (CEC2017)	10	[−100, 100]	1700
F_{22}	Hybrid function F_{14} (N = 4) (CEC2017)	10	[−100, 100]	1400
F_{23}	Hybrid function F_{20} (N = 6) (CEC2017)	10	[−100, 100]	2000
F_{24}	Hybrid function F_{11} (N = 3) (CEC2017)	10	[−100, 100]	1100
F_{25}	Hybrid function F_{6} (N = 3) (CEC 2020)	10	[−100, 100]	1700
F_{26}	Hybrid function F_{8} (N = 5) (CEC2021)	10	[−100, 100]	2100
F_{27}	Hybrid function F_{5} (N = 5) (CEC2021)	10	[−100, 100]	2100
F_{28}	Hybrid function F_{6} (N = 5) (CEC2021)	10	[−100, 100]	2200
F_{29}	Hybrid function F_{7} (N = 4) (CEC2020)	10	[−100, 100]	1600
F_{30}	Hybrid function F_{19} (N = 6) (CEC2017)	10	[−100, 100]	1900

Table 6: The description of composition functions

ID	ID in CEC	Dimension	Range	Global optimum
F_{31}	Composition function F_{20} (N = 3) (CEC 2017)	10	[−100, 100]	2100
F_{32}	Composition function F_{28} (N = 3) (CEC 2017)	10	[−100, 100]	2900
F_{33}	Composition function F_{23} (N = 4) (CEC 2017)	10	[−100, 100]	2200
F_{34}	Composition function F_{24} (N = 4) (CEC 2017)	10	[−100, 100]	2300
F_{35}	Composition function F_{8} (N = 3) (CEC 2020)	10	[−100, 100]	2300
F_{36}	Composition function F_{9} (N = 4) (CEC 2020)	10	[−100, 100]	2200
F_{37}	Composition function F_{10} (N = 5) (CEC 2020)	10	[−100, 100]	2500
F_{38}	Composition function F_{12} (N = 6) (CEC 2021)	10	[−100, 100]	2700
F_{39}	Composition function F_{11} (N = 5) (CEC 2021)	10	[−100, 100]	2600
F_{40}	Composition function F_{12} (N = 6) (CEC 2021)	10	[−100, 100]	2700

To validate the performance of the BHJO algorithm, the obtained results are compared to seven state-of-the-art optimization algorithms; namely, BWO [22], HBA [23], JS [24], WOA (Whale Optimization Algorithm) [76], MFO (Moth-Flame Optimization) [77], PSO (Particle Swarm Optimization) [14], and HHO (Harris Hawks Optimization) [78]. These algorithms are relatively new, but they have shown outstanding performance when compared to many optimizers. The population size for all the optimizers is set to 30. In addition, each algorithm was run 30 times to minimize the variance, and each run was iterated 1000 iterations to ensure the validity of the law of large numbers. The parameter settings for the algorithms utilized were obtained from their respective papers. These settings were chosen based on the authors' recommendations and empirical evaluations, ensuring consistency and comparability with the existing literature. By adopting the parameter settings from the original papers, we aimed to establish a reliable basis for evaluating and comparing the performance of our algorithm against the established state-of-the-art methods. The values used in the BHJO algorithm are taken from the three algorithms utilized for its conception. The decision to adopt specific parameter settings for the BHJO algorithm was based on extensive testing. The parameters m, f, and C were varied across three values each—−3, −2, −1 for m; 4, 5, 6 for f; and 1, 2, 3 for C–resulting in 27 unique combinations. Analysis showed that the configuration presented in Table 7 optimally balanced exploration and exploitation during the swarming process. Additionally, to maintain comparability with established research, we adopted the W_f and $\acute{C}$ values directly from the seminal paper. Despite experimenting with different population sizes–30, 50, and 100–it became evident that variations in population size did not significantly impact the performance of the hybrid algorithm. Consequently, we selected a population size of 30 for all optimizers, as this size is sufficient for the engineering design problems detailed in Section 4.2 and represents nearly the median value considered in these cases. Due to space constraints in this paper, not all experimental results could be included. By employing these specific parameter settings, we aim to ensure fair and comprehensive comparisons among the algorithms in our study, ultimately providing valuable insights into their performance and effectiveness. Table 7 provides a detailed overview of these parameters and their respective values.

Table 7: Parameters' values of the algorithms used for the comparative study

Algorithm	Parameter	Value
BHJO	m	−3
	f	5
	C	2
	W_f	$0.1 \rightarrow 0.05$
	$\acute{C}$	2
BWO	W_f	$0.1 \rightarrow 0.05$
HBA	β	6
	$\acute{C}$	2
JS	β	3
	γ	0.1
PSO	Inertia weight	$0.9 \rightarrow 0.2$
	Cognitive parameter	2
	Social parameter	2
HHO	Probability thresholds of escaping	0.5
	Escaping energy	0.5
WOA	Probability of encircling mechanism	0.5
	Spiral factor	1
MFO	Convergence constant	$-2 \rightarrow -1$
	Spiral factor	1

Tables 8 to 18 report the mean and standard deviation (STD) values averaged over 30 runs for each configuration consisting of an optimization algorithm and a test function. First, Tables 8 to 15 provide the aforementioned measurements for both unimodal and multimodal test functions across varying dimensions, specifically for dimensions 30, 50, 100, and 1000. Then, Table 16 presents the same statistical indicators for the fixed-dimension multimodal test functions across the corresponding dimensions. Finally, Tables 17 and 18 show the same statistical metrics for the hybrid and composition test functions. The standard deviation values will be utilized in various statistical tests aiming to uncover any discernible variations between the algorithms concerning their performance. In this context, it is worth emphasizing that the smaller the standard deviation value, the closer to optimal the solution obtained by the optimizer. The ideal scenario occurs when the standard deviation value equals zero, indicating that the metaheuristic algorithm has successfully obtained the global optimum when applied to the corresponding test function. Thus, the values highlighted in bold font represent the best or optimal solutions.

Table 8: Mean and STD values obtained for unimodal test functions ($D = 30$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_1	AVG	**0.00E+00**	**0.00E+00**	**1.48E−277**	2.64E−39	2.47E−01	1.64E−150	**3.05E−183**	1.00E+03
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.46E−39	1.52E−01	6.92E−150	**0.00E+00**	3.051E+03
F_2	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.26E−78	1.23E−01	**4.45E−299**	**0.00E+00**	1.66E+07
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.56E−78	2.60E−01	**0.00E+00**	**0.00E+00**	3.79E+07
F_3	AVG	**0.00E+00**	**3.94E−260**	4.68E−145	1.67E−19	4.90E+00	1.90E−101	7.06E−91	4.40E+02
	STD	**0.00E+00**	**0.00E+00**	1.99E−144	3.00E−19	2.64E+01	1.00E−100	3.87E−90	2.35E+02
F_4	AVG	**0.00E+00**	**1.77E−252**	1.83E−117	1.032E−15	1.49E+00	4.06E+01	1.00E−90	6.72E+01
	STD	**0.00E+00**	**0.00E+00**	6.94E−117	7.82E−16	2.74E−01	3.05E+01	2.31E−90	6.87E+00
F_5	AVG	**0.00E+00**	**0.00E+00**	**9.28E−274**	2.91E−38	2.64E+00	6.50E−151	**3.05E−185**	6.866E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.14E−38	1.22E+00	3.01E−150	**0.00E+00**	8.19E+04
F_6	AVG	**2.36E−223**	**0.00E+00**	**4.61E−174**	1.06E−06	7.17E+01	4.57E−07	**8.34E−187**	1.057E+03
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.62E−06	4.12E+01	1.64E−06	**0.00E+00**	1.38E+03
F_7	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.43E−163**	2.56E+00	**0.00E+00**	**0.00E+00**	2.372E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.75E+00	**0.00E+00**	**0.00E+00**	6.44E+04
Mean rank		3.29	3.29	3.29	3.79	6.86	4.36	3.29	7.86
Ranking		1	1	1	2	4	3	1	5

Table 9: Mean and STD values obtained for unimodal test functions ($D = 50$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_1	AVG	**0.00E+00**	**0.00E+00**	**2.25E−263**	2.43E−36	7.83E+00	2.46E−148	**2.80E−189**	4.03E+03
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.19E−36	2.71E+00	8.84E−148	**0.00E+00**	5.61E+03
F_2	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.551E−71	5.43E+01	**1.65E−297**	**0.00E+00**	1.09E+08
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.17E−70	3.15E+01	**0.00E+00**	**0.00E+00**	1.90E+08
F_3	AVG	**0.00E+00**	**8.52E−259**	1.55E−137	3.32E−18	2.24E+02	9.69E−98	3.71E−94	6.61E+02
	STD	**0.00E+00**	**0.00E+00**	5.49E−137	2.23E−18	1.61E+02	5.30E−97	1.82E-93	2.85E+02
F_4	AVG	**0.00E+00**	**1.22E−245**	4.31E−105	7.48E−15	3.16E+00	5.50E+01	1.09E−93	8.34E+01
	STD	**0.00E+00**	**0.00E+00**	6.94E−117	4.87E−15	3.67E−01	3.32E+01	5.29E−93	4.63E+00
F_5	AVG	**0.00E+00**	**0.00E+00**	**4.64E−259**	2.74E−35	1.49E+02	8.35E−149	**6.46E-179**	2.49E+05
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.93E−35	5.62E+01	3.78E−148	**0.00E+00**	1.82E+05
F_6	AVG	**2.98E−261**	**0.00E+00**	**4.61E−174**	3.81E−10	6.73E+02	2.76E−10	**3.11E−179**	4.23E+03
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.82E−09	1.38E+02	1.51E−09	**0.00E+00**	2.71E+03
F_7	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.84E−153	6.03E+03	**0.00E+00**	**0.00E+00**	2.39E+14
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.00E−153	8.91E+03	**0.00E+00**	**0.00E+00**	6.26E+14
Mean rank		3.29	3.29	3.29	3.79	6.86	4.36	3.29	7.86
Ranking		1	1	1	2	4	3	1	5

Table 10: Mean and STD values obtained for unimodal test functions ($D = 100$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_1	AVG	**0.00E+00**	**0.00E+00**	**2.68E−251**	6.64E−34	1.03E+02	5.91E−143	**2.80E−189**	2.89E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.56E−34	1.61E+01	3.23E−142	**0.00E+00**	1.20E+04
F_2	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.18E−67	1.03E+03	**1.24E−292**	**0.00E+00**	1.551E+09
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.05E−66	4.91E+03	**0.00E+00**	**0.00E+00**	1.41E+09
F_3	AVG	**0.00E+00**	**3.96E−258**	2.27E−132	5.61E−17	3.74E+32	1.90E−99	5.49E−94	2.37E+03
	STD	**0.00E+00**	**0.00E+00**	4.89E−132	3.33E−17	2.05E+33	8.50E−99	2.94E−93	4.42E+02
F_4	AVG	**0.00E+00**	**2.85E−245**	4.68E−82	5.00E−14	9.54E+00	7.37E+01	1.28E−92	9.35E+01
	STD	**0.00E+00**	**0.00E+00**	2.41E−81	2.67E−14	1.06E+00	2.43E+01	4.86E−92	2.05E+00
F_5	AVG	**0.00E+00**	**0.00E+00**	**9.84E−247**	2.66E−32	5.08E+03	2.86E−147	**6.46E−179**	1.70E+06
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.16E−32	1.37E+03	1.21E−146	**0.00E+00**	8.16E+05
F_6	AVG	**3.63E−274**	**0.00E+00**	**1.31E−237**	5.60E−14	1.10E+04	1.65E−28	**3.11E−179**	1.04E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.07E−13	3.37E+03	8.50E−28	**0.00E+00**	5.64E+03
F_7	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.31E−143	1.08E+07	**0.00E+00**	**0.00E+00**	8.40E+19
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.33E−142	5.91E+06	**0.00E+00**	**0.00E+00**	2.54E+19
Mean rank		3.29	3.29	3.29	4.07	7.00	4.07	3.29	7.71
Ranking		1	1	1	2	3	2	1	4

Table 11: Mean and STD values obtained for unimodal test functions ($D = 1000$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_1	AVG	**0.00E+00**	**0.00E+00**	**2.26E−229**	1.83E−30	4.43E+04	5.77E−143	**2.96E−182**	2.62E+06
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.75E−30	1.68E+03	3.06E−142	**0.00E+00**	8.99E+04
F_2	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.17E−60	1.97E+09	**1.51E−279**	**0.00E+00**	2.66E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.17E−59	1.82E+09	**0.00E+00**	**0.00E+00**	8.72E+04
F_3	AVG	**2.89E−256**	**7.16E−236**	–	–	1.00E+300	6.27E−04	1.34E−90	–
	STD	**0.00E+00**	**0.00E+00**	–	–	1.00E+300	3.57E−04	7.026E−90	–
F_4	AVG	**0.00E+00**	**0.00E+00**	**9.60E−228**	3.01E−28	8.18E+06	4.51E−145	**2.46E−177**	5.01E+08
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.05E−28	2.44E+06	2.18E−143	**0.00E+00**	1.34E+07
F_5	AVG	**0.00E+00**	**0.00E+00**	**1.39E−222**	5.94E−28	2.16E+07	6.88E−137	**2.80E−171**	1.17E+09
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.16E−27	1.09E+06	3.76E−136	**0.00E+00**	2.87E+07
F_6	AVG	**0.00E+00**	**0.00E+00**	**1.52E−228**	6.31E−32	1.53E+06	3.79E−106	**3.32E−184**	6.70E+05
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.052E−31	1.84E+05	2.08E−105	**0.00E+00**	5.10E+04
F_7	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.05E−124	8.88E+14	**0.00E+00**	**0.00E+00**	5.29E+21
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.19E−123	1.76E+15	**0.00E+00**	**0.00E+00**	2.77E+20
Mean rank		3.29	3.29	3.86	3.86	7.43	3.57	3.29	7.43
Ranking		1	1	3	3	4	2	1	4

Table 12: Mean and STD values obtained for multimodal test functions ($D = 30$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_8	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.12E+01	1.10E+01	6.48E−01	**0.00E+00**	6.896E+00
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.40E−01	6.56E−01	1.63E+00	**0.00E+00**	1.40E+00
F_9	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.37E+01	1.08E+02	**0.00E+00**	**0.00E+00**	1.57E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.57E+00	2.98E+01	**0.00E+00**	**0.00E+00**	3.90E+01
F_{10}	AVG	**4.44E−16**	**4.44E−16**	1.99E+00	5.06E−15	1.03E+00	4.11E−15	**4.44E−16**	1.630E+02

(Continued)

Table 12 (continued)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
	STD	**0.00E+00**	**0.00E+00**	6.07E+00	1.65E−15	5.65E−01	2.18E−15	**0.00E+00**	5.83E+00
F_{11}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.64E−02	2.06E−03	**0.00E+00**	1.50E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.53E−02	1.13E−02	**0.00E+00**	4.14E+01
F_{12}	AVG	**0.00E+00**	**0.00E+00**	**3.80E−279**	1.75E−40	3.75E−01	2.02E−151	**1.71E−180**	1.66E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.31E−40	2.42E−01	1.10E−150	**0.00E+00**	1.60E+02
F_{13}	AVG	**1.3498E−32**	8.98E−25	1.18E−01	9.57E−03	1.05E−01	1.52E−01	2.02E−05	2.73E+07
	STD	**5.5674E−48**	2.72E−24	1.66E−01	1.27E−02	6.74E−02	1.18E−01	2.88E−05	1.04E+08
Mean rank		2.75	2.75	4.5	4.08	6.33	4.92	3.00	4.67
Ranking		1	1	4	3	6	5	2	7

Table 13: Mean and STD values obtained for multimodal test functions ($D = 50$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_8	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.03E+01	1.94E+01	4.62E−01	**0.00E+00**	1.30E+01
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.16E−01	9.36E−01	1.86E+00	**0.00E+00**	2.04E+00
F_9	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.16E+01	3.10E+02	3.78E−15	**0.00E+00**	3.187E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.58E+01	5.98E+01	2.07E−14	**0.00E+00**	4.58E+01
F_{10}	AVG	**4.44E−16**	**4.44E−16**	3.07E+00	6.48E−15	1.03E+00	2.69E−15	**4.44E−16**	1.90E+01
	STD	**0.00E+00**	**0.00E+00**	6.07E+00	1.65E−15	3.34E−01	2.37E−15	**0.00E+00**	1.33E+00
F_{11}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.73E−01	8.19E−03	**0.00E+00**	8.806E+01
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.03E−02	2.54E−02	**0.00E+00**	7.64E+01
F_{12}	AVG	**0.00E+00**	**0.00E+00**	**3.74E−264**	2.72E−37	8.58E+00	8.13E−149	**1.86E−175**	1.890E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.65E−37	3.88E+00	2.60E−148	**0.00E+00**	2.85E+02
F_{13}	AVG	**1.3498E−32**	7.02E−25	1.29E+00	1.62E−02	2.09E+00	6.74E−01	2.17E−05	9.56E+07
	STD	**5.5674E−48**	2.78E−24	5.86E−01	1.88E−02	7.50E−01	3.44E−01	2.505E−05	3.35E+08
Mean rank		2.67	2.67	4.25	4.08	6.83	4.92	2.92	7.67
Ranking		1	1	4	3	6	5	2	7

Table 14: Mean and STD values obtained for multimodal test functions ($D = 100$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_8	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.37E+01	4.19E+01	1.04E−01	**0.00E+00**	3.27E+01
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.29E−01	1.07E−01	5.73E−01	**0.00E+00**	1.47E+00
F_9	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.06E+03	**0.00E+00**	**0.00E+00**	7.66E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.35E+02	**0.00E+00**	**0.00E+00**	7.02E+01
F_{10}	AVG	**4.44E−16**	**4.44E−16**	6.62E−01	6.48E−15	5.38E+00	2.69E−15	**4.44E−16**	1.98E+01
	STD	**0.00E+00**	**0.00E+00**	3.63E+00	1.6559E−15	3.338E−01	2.3756E−15	**0.00E+00**	2.134E−01
F_{11}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.20E−01	**0.00E+00**	**0.00E+00**	2.606E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.11E−02	**0.00E+00**	**0.00E+00**	1.01E+02
F_{12}	AVG	**0.00E+00**	**0.00E+00**	**9.70E−252**	4.04E−35	1.06E+02	5.39E−148	**1.86E−175**	1.65E+03
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.82E−35	1.93E+01	2.66E−147	**0.00E+00**	6.74E+02
F_{13}	AVG	**1.3498E−32**	4.30E−26	7.41E+00	1.86E−01	1.03E+02	1.59E+00	2.84E−05	3.70E+08
	STD	**5.5674E−48**	1.46E−25	6.56E−01	7.84E−02	3.29E+01	8.06E−01	4.52E−05	3.45E+08
Mean rank		2.92	2.92	4.33	4.08	6.83	4.25	3.17	7.50
Ranking		1	1	5	3	6	4	2	7

Table 15: Mean and STD values obtained for multimodal test functions ($D = 1000$)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_8	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.20E+02	4.60E+02	9.44E−01	**0.00E+00**	4.67E+02
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.11E+02	3.97E+04	5.17E+00	**0.00E+00**	3.42E+00
F_9	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**3.20E+02**	1.62E+04	6.06E−14	**0.00E+00**	1.47E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.33E+02	3.32E−13	**0.00E+00**	1.89E+02
F_{10}	AVG	**4.44E−16**	**4.44E−16**	2.65E+00	**7.54E−15**	1.61E+01	4.20E−15	**4.44E−16**	2.01E+01
	STD	**0.00E+00**	**0.00E+00**	6.88E+00	**0.00E+00**	3.51E+02	2.45E−15	**0.00E+00**	1.96E−01
F_{11}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.55E−17	3.044E+01	**0.00E+00**	**0.00E+00**	2.22E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.64E−17	1.93E+00	**0.00E+00**	**0.00E+00**	5.33E+02
F_{12}	AVG	**0.00E+00**	**0.00E+00**	**1.60E−232**	4.92E−32	2.38E+04	6.15E−146	1.25E−182	3.37E+04
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.51E−32	1.58E+03	3.21E−145	**0.00E+00**	5.29E+03
F_{13}	AVG	5.51E−04	**7.97E−26**	9.91E+01	6.66E+00	1.52E+08	2.09E+01	1.27E−04	4.94E+10
	STD	9.75E−04	**2.18E−25**	2.42E−01	1.97E+00	1.60E+07	5.10E+00	1.21E−04	1.52E+09
Mean rank		3.08	2.75	3.92	4.17	7.50	4.67	2.92	7.00
Ranking		3	1	4	5	8	6	2	7

Table 16: Mean and STD values obtained for fixed-dimension multimodal test functions

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_{14}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**3.36E−165**	3.53E−31	**0.00E+00**	**0.00E+00**	1.14E−48
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.35E−30	**0.00E+00**	**0.00E+00**	6.23E−48
F_{15}	AVG	3.37E−04	3.44E−04	3.36E−04	**3.07E−04**	9.16E−04	2.78E−04	9.79E−04	4.53E−04
	STD	2.3359E−05	5.0034E−05	7.46E−04	**9.30E−07**	2.41E−04	2.78E−04	3.70E−04	1.03E−04
F_{16}	AVG	−2.00E+03	−1.55E+03	−1.00E+03	−9.98E+02	−1.00E+03	−1.99E+03	**−1.00E+03**	−1.87E+03
	STD	1.43E−08	2.84E+02	1.15E−13	3.29E+00	3.58E−13	1.90E+01	**0.00E+00**	2.01E+02
F_{17}	AVG	**0.00E+00**	5.83E−33	5.55E−15	2.51E−12	1.50E−10	6.44E−06	5.71E−11	5.93E−07
	STD	**0.00E+00**	3.08E−32	1.16E−14	4.92E−12	5.09E−10	1.23E−05	3.00E−10	2.11E−06
F_{18}	AVG	8.44E−04	**3.37E−04**	5.41E−03	3.15E−04	9.59E−04	7.17E−04	3.26E−04	1.65E−03
	STD	3.5079E−04	**4.40E−05**	9.11E−03	4.17E−05	2.14E−04	4.01E−04	1.85E−05	3.56E−03
F_{19}	AVG	**8.8401E−18**	1.26E−03	6.12E−01	2.02E−06	4.03E−02	1.20E+00	8.67E−05	1.12E+00
	STD	**2.6898E−17**	1.081E−03	1.39E+00	6.70E−01	5.79E−02	1.66E+00	1.23E-04	1.17E−03
F_{20}	AVG	**0.00E+00**	**0.00E+00**	**0.00E+00**	**5.13E−244**	1.49E−38	1.58E−215	4.13E−215	8.58E−212
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.20E−38	**0.00E+00**	**0.00E+00**	**0.00E+00**
Mean rank		3.21	3.93	5.29	3.86	4.57	6.14	3.57	5.43
Ranking		1	4	6	3	5	8	2	7

Table 17: Mean and STD values obtained for hybrid test functions

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_{21}	AVG	**1.77E+03**	1.80E+03	1.74E+03	1.72E+03	1.77E+03	1.81E+03	1.77E+03	1.79E+03
	STD	**7.70E+00**	2.02E+01	2.61E+01	1.20E+01	4.74E+01	5.05E+01	3.48E+01	5.90E+01
F_{22}	AVG	1.54E+03	1.64E+03	2.49E+03	**1.45E+03**	1.82E+03	2.38E+03	1.78E+03	5.60E+03
	STD	3.92E+01	6.49E+01	4.99E+03	**2.65E+01**	7.76E+02	1.27E+03	5.89E+02	5.80E+03
F_{23}	AVG	2.18E+03	2.20E+03	2.08E+03	**2.02E+03**	2.09E+03	2.16E+03	2.18E+03	2.08E+03
	STD	4.15E+01	4.72E+01	8.01E+01	**1.05E+01**	6.22E+01	7.01E+01	6.94E+01	7.51E+02
F_{24}	AVG	1.19E+03	1.78E+03	1.11E+03	**1.11E+03**	1.13E+03	1.20E+03	1.15E+03	1.26E+03
	STD	4.79E+01	3.21E+02	1.18E+01	**6.73E+00**	2.54E+01	8.11E+01	3.44E+01	3.46E+02

(Continued)

Table 17 (continued)

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_{25}	AVG	**4.61E+01**	5.35E+01	1.08E+02	1.13E+01	5.76E+01	1.36E+02	5.35E+01	1.28E+02
	STD	**5.40E+00**	8.89E+00	1.23E+02	9.60E+00	7.77E+01	1.08E+02	6.06E+01	1.47E+02
F_{26}	AVG	**2.230E+03**	2.23E+03	2.22E+03	2.82E+03	2.21E+03	2.22E+03	2.23E+03	2.22E+03
	STD	**1.59E+00**	3.33E+00	2.82E+01	9.58E+00	1.68E+01	1.28E+01	1.76E+01	1.00E+01
F_{27}	AVG	2.04E+03	2.09E+03	2.02E+03	**2.02E+03**	2.04E+03	2.05E+03	2.07E+03	2.04E+03
	STD	8.40E+00	1.90E+01	8.01E+00	**6.43E+00**	3.27E+01	2.33E+01	3.60E+01	2.51E+01
F_{28}	AVG	**2.04E+03**	2.09E+03	2.02E+03	2.01E+03	2.04E+03	2.05E+03	2.11E+03	2.04E+03
	STD	**6.82E+00**	1.94E+01	1.09E+01	7.14E+00	3.21E+01	2.13E+01	5.68E+01	2.55E+01
F_{29}	AVG	5.39E+03	1.13E+04	**7.60E+02**	1.26E+03	4.54E+03	2.98E+04	6.21E+03	5.71E+04
	STD	3.07E+03	7.81E+03	**3.17E+02**	1.29E+03	4.98E+03	8.53E+04	5.26E+03	2.58E+04
F_{30}	AVG	5.43E+03	1.02E+04	3.02E+03	**1.93E+03**	3.16E+03	3.69E+04	8.83E+03	1.61E+04
	STD	3.91E+03	2.73E+03	5.67E+03	**2.74E+01**	1.80E+03	6.73E+03	6.72E+03	2.16E+04
Mean rank		2.30	3.70	4.60	1.70	4.90	6.10	5.60	7.10
Ranking		2	3	4	1	5	7	6	8

Table 18: Mean and STD values obtained for composition test functions

ID		BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
F_{31}	AVG	**2.20E+03**	2.25E+03	2.29E+03	2.25E+03	2.32E+03	2.32E+03	2.30E+03	2.32E+03
	STD	**3.04E+00**	1.63E+01	5.03E+01	5.36E+01	5.49E+01	5.11E+01	7.19E+01	4.29E+01
F_{32}	AVG	**3.27E+03**	3.54E+03	3.39E+03	3.16E+03	3.17E+03	3.42E+03	3.36E+03	3.34E+03
	STD	**3.36E+01**	9.72E+01	2.24E+02	1.06E+02	5.40E+01	1.59E+02	1.11E+02	1.02E+03
F_{33}	AVG	2.32E+03	2.56E+03	**2.30E+03**	2.29E+03	2.38E+03	2.35E+03	2.51E+03	2.31E+03
	STD	6.47E+00	1.98E+02	**1.19E+00**	1.59E+01	2.86E+02	2.29E+02	4.71E+02	2.08E+02
F_{34}	AVG	**2.53E+03**	2.69E+03	2.75E+03	2.66E+03	2.78E+03	2.77E+03	2.82E+03	2.76E+03
	STD	**1.12E+01**	7.76E+01	2.67E+01	1.10E+02	9.60E+01	5.01E+01	4.39E+01	5.03E+01
F_{35}	AVG	1.50E+02	1.50E+02	**1.50E+02**	1.20E+02	1.37E+02	1.50E+02	1.50E+02	1.24E+02
	STD	5.32E−09	5.88E−09	**3.03E−14**	5.55E+01	3.13E+01	7.91E−10	3.26E−08	4.82E+01
F_{36}	AVG	**2.00E+02**	**2.00E+02**	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02	**2.00E+02**	2.00E+02
	STD	**2.89E−14**	**2.89E−14**	**2.89E−14**	4.25E−14	2.82E−12	3.98E−14	**2.89E−14**	1.58E−14
F_{37}	AVG	2.50E+03	**2.50E+03**	**2.50E+03**	**2.50E+03**	2.50E+03	2.50E+03	02.50E+03	2.50E+03
	STD	2.04E−14	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.79E−13	2.89E−14	2.30E−14	1.49E−14
F_{38}	AVG	2.70E+03	2.701E+03	2.70E+03	2.70E+03	**2.70E+03**	2.70E+03	2.70E+03	2.70E+03
	STD	6.35E−01	3.18E+01	5.46E−01	6.02E−00	**6.22E−01**	7.73E−01	7.12E−01	8.52E+00
F_{39}	AVG	**2.60E+03**	**2.60E+03**	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03
	STD	**9.49E−05**	**9.49E−05**	2.85E−04	2.30E−04	2.34E−04	1.45E−04	1.14E−04	2.02E−04
F_{40}	AVG	**1.78E+03**	1.81E+03	1.74E+03	**1.72E+03**	1.75E+03	1.79E+03	1.78E+03	1.76E+03
	STD	**1.07E+01**	1.85E+01	3.21E+01	**1.070E+01**	3.30E+01	4.87E+01	3.84E+01	3.65E+02
Mean rank		2.25	3.70	3.35	5.10	5.90	5.25	5.05	5.40
Ranking		1	3	2	5	8	6	4	7

Then, we performed eleven Friedman tests using the standard deviation values reported in Tables 8 to 18, with a significance level of 0.05. The p-values computed by the Friedman test are reported in Table 19. It is important to note that if the p-value is equal to or less than 0.05, it is concluded that there is a significant difference in performance between the algorithms of the comparative study. Conversely, if the p-value is greater than 0.05, it indicates that all the algorithms exhibit similar performance.

Upon observation, we find that all the p-values, except for experiment 9, are less than or equal to 0.05, indicating a difference in performance among the algorithms. Therefore, we will proceed to conduct the post-hoc Dunn's test for these cases to analyze their performance more comprehensively.

Table 19: Hypothesis test summary

	Experiment	p-value	Decision
1	Table 8	6.07E−7	The test reveals a difference in performance among the algorithms.
2	Table 9	6.07E−7	The test reveals a difference in performance among the algorithms.
3	Table 10	8.38E−7	The test reveals a difference in performance among the algorithms.
4	Table 11	3.14E−7	The test reveals a difference in performance among the algorithms.
5	Table 12	2.20E−4	The test reveals a difference in performance among the algorithms.
6	Table 13	6.20E−5	The test reveals a difference in performance among the algorithms.
7	Table 14	1.20E−4	The test reveals a difference in performance among the algorithms.
8	Table 15	7.40E−5	The test reveals a difference in performance among the algorithms.
9	Table 16	9.50E−2	The test does not reveal a difference in performance among the algorithms.
10	Table 17	1.00E−6	The test reveals a difference in performance among the algorithms.
11	Table 18	8.70E−3	The test reveals a difference in performance among the algorithms.

Upon examining the last and penultimate lines of Tables 8 to 18, presenting the mean ranks and the rankings respectively, a distinct pattern emerges regarding the performance of the BHJO algorithm. Notably, the BHJO algorithm achieves the highest score in the first position for a remarkable nine out of the eleven cases studied. In addition, it secures the second position in one case and the third position in another case. Specifically:

- First, the BHJO algorithm excels in cases dedicated to analyzing unimodal test functions, consistently ranking first. This remarkable performance highlights its exceptional exploitation capability, establishing it as the top-performing algorithm among those considered in the comparative study.
- Furthermore, in the context of multimodal test functions, the BHJO algorithm secures the first position in four out of five cases and the third position in one case. The case where the BHJO algorithm ranks third will be further investigated in the post-hoc Dunn's test.
- Finally, when confronted with hybrid test functions, the algorithm ranks second, and when faced with composition test functions, it emerges as the frontrunner, securing the first position. The case in which the BHJO algorithm attains a second-place ranking will undergo additional investigation in the post-hoc Dunn's test.

The post-hoc Dunn's test, also known as Dunn's multiple comparison tests, is a statistical test used to compare multiple groups or conditions after performing a non-parametric test, such as the Friedman test or the Kruskal-Wallis test. It allows for pairwise comparisons between the groups to determine if there are significant differences between them. Dunn's test utilizes rank-based methods and applies appropriate adjustments for multiple comparisons, such as the Bonferroni correction or the Holm-Bonferroni method. This test helps to identify specific group differences and provides further insights into the significance of those differences. Tables 20 to 29 report the significance values that have been adjusted by the Bonferroni correction for multiple tests. In this analysis, the significance of differences between algorithms is indicated by the p-values, with bold font highlighting cases where the algorithm in the row demonstrates superior performance compared to that in the corresponding column. On the other hand, *p*-values presented in normal font signify instances where both algorithms exhibit similar performance.

Table 20: Results of Dunn's test using STD values reported in Table 8

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
BWO	–	–	1.00E+00	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
HBA	–	–	–	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
JS	–	–	–	–	5.30E−01	1.00E+00	–	**5.00E−02**
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	–	–	1.00E+00	–	–	2.10E−01
HHO	–	–	–	1.00E+00	1.80E−01	1.00E+00	–	**1.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 21: Results of Dunn's test using STD values reported in Table 9

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
BWO	–	–	1.00E+00	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
HBA	–	–	–	1.00E+00	1.80E−01	1.00E+00	1.00E+00	**1.00E−02**
JS	–	–	–	–	5.30E−01	1.00E+00	–	**5.00E−02**
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	–	–	1.00E+00	–	–	2.10E−01
HHO	–	–	–	1.00E+00	1.80E−01	1.00E+00	–	**1.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 22: Results of Dunn's test using STD values reported in Table 10

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	1.30E−01	1.00E+00	1.00E+00	**2.00E−02**
BWO	–	–	1.00E+00	1.00E+00	1.30E−01	1.00E+00	1.00E+00	**2.00E−02**
HBA	–	–	–	1.00E+00	1.30E−01	1.00E+00	1.00E+00	**2.00E−02**
JS	–	–	–	–	7.10E−01	1.00E+00	–	1.50E−01
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	–	–	7.10E−01	–	–	1.50E−01
HHO	–	–	–	1.00E+00	1.30E−01	1.00E+00	–	**2.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 23: Results of Dunn's test using STD values reported in Table 11

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	**4.00E−02**	1.00E+00	1.00E+00	**4.00E−02**
BWO	–	–	1.00E+00	1.00E+00	**4.00E−02**	1.00E+00	1.00E+00	**4.00E−02**
HBA	–	–	–	1.00E+00	1.80E−01	–	–	1.80E−01
JS	–	–	–	–	1.80E−01	–	–	1.80E−01
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	1.00E+00	1.00E+00	9.00E−02	–	–	9.00E−02
HHO	–	–	1.00E+00	1.00E+00	**4.00E−02**	1.00E+00	–	**4.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 24: Results of Dunn's test using STD values reported in Table 12

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	3.20E−01	1.00E+00	1.00E+00	**1.00E−02**
BWO	–	–	1.00E+00	1.00E+00	3.20E−01	1.00E+00	1.00E+00	**1.00E−02**
HBA	–	–	–	–	1.00E+00	1.00E+00	–	7.00E−01
JS	–	–	1.00E+00	–	1.00E+00	1.00E+00	–	3.20E−01
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	–	–	1.00E+00	–	–	1.00E+00
HHO	–	–	1.00E+00	1.00E+00	5.20E−01	1.00E+00	–	**3.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 25: Results of Dunn's test using STD values reported in Table 13

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	9.00E−02	1.00E+00	1.00E+00	**1.00E−02**
BWO	–	–	1.00E+00	1.00E+00	9.00E−02	1.00E+00	1.00E+00	**1.00E−02**
HBA	–	–	–	–	1.00E+00	1.00E+00	–	4.40E−01
JS	–	–	1.00E+00	–	1.00E+00	1.00E+00	–	3.20E−01
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	–	–	1.00E+00	–	–	1.00E+00
HHO	–	–	1.00E+00	1.00E+00	1.60E−01	1.00E+00	–	**2.00E−02**
MFO	–	–	–	–	–	–	–	–

Table 26: Results of Dunn's test using STD values reported in Table 14

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	1.00E+00	1.60E−01	1.00E+00	1.00E+00	**3.00E−02**
BWO	–	–	1.00E+00	1.00E+00	1.60E−01	1.00E+00	1.00E+00	**3.00E−02**
HBA	–	–	–	–	1.00E+00	–	–	7.00E−01
JS	–	–	1.00E+00	–	1.00E+00	1.00E+00	–	4.40E−01
PSO	–	–	–	–	–	–	–	1.00E+00
WOA	–	–	1.00E+00	–	1.00E+00	–	–	6.00E−01
HHO	–	–	1.00E+00	1.00E+00	2.70E−01	1.00E+00	–	6.00E−02
MFO	–	–	–	–	–	–	–	–

Table 27: Results of Dunn's test using STD values reported in Table 15

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	–	1.00E+00	1.00E+00	**5.00E−02**	1.00E+00	–	1.60E−01
BWO	1.00E+00	–	1.00E+00	1.00E+00	**2.00E−02**	1.00E+00	1.00E+00	7.00E−02
HBA	–	–	–	1.00E+00	3.20E−01	1.00E+00	–	8.20E−01
JS	–	–	–	–	5.20E−01	1.00E+00	–	1.00E+00
PSO	–	–	–	–	–	–	–	–
WOA	–	–	–	–	1.00E+00	–	–	1.00E+00
HHO	1.00E+00	–	1.00E+00	1.00E+00	**3.00E−02**	1.00E+00	–	1.10E−01
MFO	–	–	–	–	1.00E+00	–	–	–

Table 28: Results of Dunn's test using STD values reported in Table 17

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	–	4.90E−01	**1.00E−02**	7.00E−02	**3.30E−04**
BWO	–	–	1.00E+00	–	1.00E+00	8.00E−01	1.00E+00	**5.00E−02**
HBA	–	–	–	–	1.00E+00	1.00E+00	1.00E+00	6.30E−01
JS	1.00E+00	1.00E+00	2.30E−01	–	1.00E−01	**0.00E+00**	**1.00E−02**	**2.31E−05**
PSO	–	–	–	–	–	1.00E+00	1.00E+00	1.00E+00
WOA	–	–	–	–	–	–	–	1.00E+00
HHO	–	–	–	–	–	1.00E+00	–	1.00E+00
MFO	–	–	–	–	–	–	–	–

Table 29: Results of Dunn's test using STD values reported in Table 18

	BHJO	BWO	HBA	JS	PSO	WOA	HHO	MFO
BHJO	–	1.00E+00	1.00E+00	2.60E−01	**2.00E−02**	1.70E−01	3.00E−01	1.10E−01
BWO	–	–	–	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
HBA	–	1.00E+00	–	1.00E+00	5.60E−01	1.00E+00	1.00E+00	1.00E+00
JS	–	–	–	–	1.00E+00	1.00E+00	–	1.00E+00
PSO	–	–	–	–	–	–	–	–
WOA	–	–	–	–	1.00E+00	–	–	1.00E+00
HHO	–	–	–	1.00E+00	1.00E+00	1.00E+00	–	1.00E+00
MFO	–	–	–	–	1.00E+00	–	–	–

Now, let us consider the cases previously mentioned where the BHJO algorithm did not secure the first rank. Referring to Table 15, it becomes evident that the BHJO algorithm occupies the third position, trailing behind the BWO and HHO algorithms, respectively. In Table 27, the p-values are both 1, indicating that the BHJO, BWO, and HHO algorithms demonstrate similar performance. Shifting our focus to Tables 17, the BHJO algorithm is seen as the second-best, following the JS algorithm. Furthermore, Table 28 reveals a p-value of 1, suggesting comparable performance between the BHJO and JS algorithms.

4.2 Performance of the BHJO Algorithm on COPs

This section focuses on evaluating the effectiveness of the BHJO algorithm in tackling a range of 33 distinct engineering design problems [79]. The specific details and characteristics of these constrained optimization problems can be found in Tables 30–32. Furthermore, to assess the performance of the BHJO algorithm, a comparative analysis is conducted against five optimization algorithms, denoted as BWO [22], HBA [23], the Improved Unified Differential Evolution (IUDE) [80], the Matrix Adaptation Evolution Strategy (ϵ MAgES) [81], and the Improved Success-History based Adaptive Differential Evolution with Linear Population Size Reduction and ϵ-level control (iLSHADE $_\epsilon$) [82]. For each combination of algorithm and design problem, the experiments are repeated 25 times, and the resulting statistical measures, including minimum, maximum, median, mean, and standard deviation values, are documented. Tables 33–38 summarize these comprehensive statistical indicators, providing valuable insights into the relative performance of the algorithms across the evaluated design problems. In addition, we calculate three other performance metrics [79]:

1. The Feasibility Rate (FR), which represents the ratio of runs where at least one feasible solution is achieved within the maximum function evaluations out of the total number of runs.
2. The Mean constraint Violation (MV), which is computed using Eq. (20).

$$MV = \frac{\sum_{i=1}^{M} \max\left(g_i\left(X\right),\ 0\right) + \sum_{j=1}^{N} \max\left(\left|h_j\left(X\right)\right| - 10^{-4},\ 0\right)}{M+N} \tag{20}$$

3. The Success Rate (SR), which indicates the ratio of runs where an algorithm successfully obtains a feasible solution X that satisfies the constrained $f\left(X\right) - f\left(X^*\right) \leq 10^{-8}$ within the maximum function evaluations out of all the runs performed.

Table 30: Details of the industrial chemical processes. D, g, and h denote respectively the number of decision variables, inequality constraints, and equality constraints. The last column (i.e., $f(X^*)$) represents the best known optimal solution [79]

ID	Name and attributes: [D, g, h ,$f(X^*)$]
Pr01	Heat exchanger network design (case 1): [9,0,8,189.31162966]
Pr02	Heat exchanger network design (case 2): [11,0,9,7049.0369540]
Pr03	Optimal operation of alkylation unit: [7,14,0, −4529.1197395]
Pr04	Reactor network design: [6,1,4, −0.38826043623]
Pr05	Haverly's pooling problem: [9,2,4, −400.00560000]
Pr06	Blending-pooling-separation problem: [38,0,32,1.8638304088]
Pr07	Propane, isobutane, n-butane nonsharp separation: [48,0,38,2.1158627569]

Table 31: Details of the process synthesis and design problems. D, g, and h denote respectively the number of decision variables, inequality constraints and equality constraints. The last column (i.e., $f(X^*)$) represents the best known optimal solution [79]

ID	Name and attributes: [D, g, h, $f(X^*)$]
Pr08	Process synthesis problem: [2,2,0,2.0000000000]
Pr09	Process synthesis and design problem: [3,1,1,2.5576545740]
Pr10	Process flow sheeting problem: [3,3,0,1.0765430833]
Pr11	Two-reactor problem: [7,4,4,99.238463653]
Pr08	Process synthesis problem: [7,9,0,2.9248305537]
Pr13	Process design problem: [5,3,0,26887.000000]
Pr14	Multi-product batch plant: [10,10,0,53638.942722]

Table 32: Details of the mechanical engineering problems. D, g, and h denote respectively the number of decision variables, inequality constraints and equality constraints. The last column (i.e., $f(X^*)$) represents the best known optimal solution [79]

ID	Name and attributes: [D, g, h, $f(X^*)$]
Pr15	Weight minimization of a speed reducer: [7,11,0,2.9944244658E+03]
Pr16	Optimal design of industrial refrigeration system: [14,15,0,3.2213000814E−02]
Pr17	Tension/Compression spring design (case 1: [3,3,0,1.2665232788E−02])
Pr18	Pressure vessel design: [4,4,0,5885.3327736]
Pr19	Welded beam design: [4,5,0,1.6702177263]
Pr20	Three-bar truss design problem: [2,3,0,263.89584338E]
Pr21	Multiple disk clutch brake design problem: [5,7,0,0.23524245790]
Pr22	Planetary gear train design optimization problem: [9,10,1,0.52576870748]
Pr23	Step-cone pulley problem: [5,8,3,16.069868725]

(Continued)

Table 32 (continued)

ID	Name and attributes: $[D, g, h, f(X^*)]$
Pr24	Robot gripper problem: [7,7,0,2.5287918415]
Pr25	Hydro-static thrust bearing design problem: [4,7,0,1625.4428092]
Rr26	Four-stage gear box problem: [22,86,0,35.359231973]
Pr27	10-Bar truss design: [10,3,0,524.45076066]
Pr28	Rolling element bearing: [10,9,0,14614.135715]
Pr29	Gas transmission compressor design (GTCD): [4,1,0,2964895.4173]
Pr30	Tension/Compression spring design (case 2): [3,8,0,2.6138840583]
Pr31	Gear train design problem: [4,1,1,0.00]
Pr32	Himmelblau's function: [5,6,0, −30665.538672]
Pr33	Topology optimization: [30,30,0,2.6393464970]

Table 33: Statistical results of the BHJO algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	189.0	189.0	189.0	189.0	0.158	59	112000.0	44
Pr02	7050.0	7050.0	7050.0	7050.0	0.535	98	6170.0	99
Pr03	−4530.0	−4530.0	−4530.0	−4530.0	0.501	68	4.69	23
Pr04	−0.377	−0.322	−0.329	−0.286	0.0668	40	0.811	72
Pr05	−400.0	−400.0	−400.0	−400.0	0.00296	100	−0.0928	96
Pr06	1.71	1.78	1.79	1.86	0.0937	99	1.45	5
Pr07	1.77	1.98	1.96	2.1	0.186	89	0.101	84
Pr08	2.0	2.0	2.0	2.0	0.0	100	0.276	100
Pr09	2.56	2.56	2.56	2.56	0.00112	100	0.629	100
Pr10	1.08	1.08	1.08	1.08	0.00208	100	−0.403	88
Pr11	99.2	99.2	99.2	99.2	0.025	100	2.05	79
Pr12	2.92	2.92	2.92	2.92	0.00288	100	−1.07	86
Pr13	26900.0	26900.0	26900.0	26900.0	7.31	100	−0.0142	100
Pr14	53700.0	56100.0	55800.0	58300.0	2500.0	100	−0.4	98
Pr15	2990.0	2990.0	2990.0	2990.0	2.91	100	−1.44	100
Pr16	0.0322	0.0322	0.0322	0.0322	5.8E−6	81	−0.422	94
Pr17	0.0127	0.0127	0.0127	0.0127	1.97E−5	100	0.4	100
Pr18	5900.0	5970.0	5970.0	6050.0	94.4	100	−1.67	90
Pr19	1.67	1.67	1.67	1.67	7.78E−5	100	0.737	100
Pr20	264.0	264.0	264.0	264.0	0.058	100	1.26	100
Pr21	0.235	0.235	0.235	0.235	1.54E−4	100	−3.13	100
Pr22	0.526	0.526	0.526	0.526	1.4E−4	100	−0.0339	81
Pr23	16.1	16.1	16.1	16.1	0.0202	100	−1.15	100
Pr24	2.53	2.53	2.53	2.54	0.00664	100	−0.566	100
Pr25	1630.0	1740.0	1740.0	1860.0	133.0	44	0.673	91
Pr26	35.6	41.1	40.7	45.3	6.11	76	1.52	73

(Continued)

Table 33 (continued)

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr27	524.0	524.0	524.0	524.0	0.26	100	−0.276	98
Pr28	14600.0	14600.0	14600.0	14600.0	8.44	100	1.35	100
Pr29	2960000.0	2960000.0	2960000.0	2960000.0	2750.0	100	−0.844	100
Pr30	2.61	2.64	2.64	2.66	0.0267	98	0.781	91
Pr31	0.0	0.0	0.0	0.0	0.0	100	0.0692	100
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	21.0	100	1.05	100
Pr33	2.64	2.64	2.64	2.64	3.81E−4	100	1.18	100

Table 34: Statistical results of the BWO algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	189.0	189.0	189.0	189.0	0.204	57	112000.0	83
Pr02	7050.0	7050.0	7050.0	7050.0	0.585	95	6170.0	98
Pr03	−4530.0	−4530.0	−4530.0	−4530.0	0.557	65	3.51	66
Pr04	−0.38	−0.358	−0.347	−0.295	0.051	36	−0.418	75
Pr05	−400.0	−400.0	−400.0	−400.0	0.00331	100	−0.097	93
Pr06	1.71	1.83	1.81	1.86	0.0707	65	1.44	47
Pr07	1.77	1.95	1.93	2.09	0.205	74	−0.126	83
Pr08	2.0	2.0	2.0	2.0	0.0	100	−1.58	100
Pr09	2.56	2.56	2.56	2.56	0.00135	100	0.425	100
Pr10	1.08	1.08	1.08	1.08	0.00199	100	0.326	94
Pr11	99.2	99.2	99.2	99.2	0.0215	100	−0.0299	57
Pr12	2.92	2.92	2.92	2.92	0.00214	100	0.551	51
Pr13	26900.0	26900.0	26900.0	26900.0	8.39	100	1.06	100
Pr14	53700.0	56100.0	56000.0	58300.0	2700.0	100	−0.453	63
Pr15	2990.0	2990.0	2990.0	2990.0	2.4	100	0.881	100
Pr16	0.0322	0.0322	0.0322	0.0322	7.73E−6	85	−0.096	81
Pr17	0.0127	0.0127	0.0127	0.0127	1.96E−5	100	−0.505	100
Pr18	5890.0	6000.0	5980.0	6060.0	111.0	100	−1.02	67
Pr19	1.67	1.67	1.67	1.67	1.31E−4	100	−1.25	100
Pr20	264.0	264.0	264.0	264.0	0.0481	100	−0.307	100
Pr21	0.235	0.235	0.235	0.235	1.31E−4	100	0.548	100
Pr22	0.526	0.526	0.526	0.526	1.39E−4	100	−0.798	99
Pr23	16.1	16.1	16.1	16.1	0.0186	100	−1.42	100
Pr24	2.53	2.53	2.53	2.54	0.00628	100	−0.736	100
Pr25	1650.0	1740.0	1740.0	1850.0	127.0	44	−1.32	79
Pr26	35.6	41.3	40.3	44.3	5.65	38	−0.627	96
Pr27	524.0	524.0	524.0	524.0	0.27	100	0.193	95
Pr28	14600.0	14600.0	14600.0	14600.0	6.77	100	0.106	100
Pr29	2960000.0	2960000.0	2960000.0	2960000.0	2770.0	100	0.481	100

(Continued)

Table 34 (continued)

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
RCy30	2.62	2.64	2.64	2.66	0.0284	93	0.566	81
Pr31	0.0	0.0	0.0	0.0	0.0	100	−1.26	100
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	19.0	100	−1.87	100
Pr33	2.64	2.64	2.64	2.64	4.06E−4	100	−0.622	100

Table 35: Statistical results of the HBA algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	189.0	189.0	189.0	189.0	0.153	41	112000.0	53
Pr02	7050.0	7050.0	7050.0	7050.0	0.54	98	6170.0	97
Pr03	−4530.0	−4530.0	−4530.0	−4530.0	0.479	95	5.2	24
RCy04	−0.387	−0.343	−0.341	−0.289	0.0547	65	0.485	56
Pr05	−400.0	−400.0	−400.0	−400.0	0.00344	100	−2.23	99
Pr06	1.71	1.76	1.77	1.85	0.101	52	2.0	9
Pr07	1.79	1.97	1.94	2.11	0.198	20	−1.17	94
Pr08	2.0	2.0	2.0	2.0	0.0	100	1.42	100
Pr09	2.56	2.56	2.56	2.56	0.00122	100	−0.464	100
Pr10	1.08	1.08	1.08	1.08	0.00204	100	0.317	90
Pr11	99.2	99.2	99.2	99.2	0.0233	100	−1.67	84
Pr12	2.92	2.92	2.92	2.92	0.00269	100	−2.81	59
Pr13	26900.0	26900.0	26900.0	26900.0	7.39	100	−0.87	100
Pr14	53800.0	56000.0	56000.0	58500.0	2780.0	100	0.232	33
Pr15	2990.0	2990.0	2990.0	2990.0	2.5	100	0.605	100
Pr16	0.0322	0.0322	0.0322	0.0322	8.08E−6	92	1.85	89
Pr17	0.0127	0.0127	0.0127	0.0127	2.14E−5	100	−0.717	100
Pr18	5890.0	5970.0	5970.0	6050.0	92.4	100	0.508	79
Pr19	1.67	1.67	1.67	1.67	1.1E−4	100	0.55	100
Pr20	264.0	264.0	264.0	264.0	0.0549	100	0.421	100
Pr21	0.235	0.235	0.235	0.235	1.48E−4	100	−0.745	100
Pr22	0.526	0.526	0.526	0.526	1.46E−4	100	0.593	37
RCy23	16.1	16.1	16.1	16.1	0.0173	100	−1.26	100
Pr24	2.53	2.54	2.54	2.54	0.00684	100	−0.247	100
Pr25	1630.0	1760.0	1750.0	1850.0	143.0	93	−0.146	93
Pr26	35.5	40.7	40.0	44.8	5.51	58	0.757	3
Pr27	524.0	524.0	524.0	524.0	0.274	100	−1.09	96
Pr28	14600.0	14600.0	14600.0	14600.0	7.07	100	1.62	100
Pr29	2960000.0	2960000.0	2960000.0	2960000.0	2410.0	100	−0.323	100
RCy30	2.62	2.64	2.64	2.66	0.0233	97	0.762	50
Pr31	0.0	0.0	0.0	0.0	0.0	100	1.13	100

(Continued)

Table 35 (continued)

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	17.3	100	−0.169	100
Pr33	2.64	2.64	2.64	2.64	3.76E−4	100	−0.469	100

Table 36: Statistical results of the IUDE algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	189.0	260.0	229.0	185.0	80.6	24	112000.0	4
Pr02	7050.0	7050.0	7150.0	5940.0	754.0	92	6170.0	92
Pr03	−4530.0	−143.0	−6250.0	−18300.0	6750.0	64	3.66	12
Pr04	−0.286	−0.592	−0.496	−1.0	0.182	0	0.079	0
Pr05	−400.0	−400.0	−351.0	−0.0083	132.0	100	0.0	80
Pr06	1.71	0.998	1.07	0.998	0.198	0	2.1	0
Pr07	1.76	1.34	1.4	0.998	0.382	0	0.282	0
Pr08	2.0	2.0	2.0	2.0	6.41E−17	100	0.0	100
Pr09	2.56	2.56	2.56	2.56	1.36E−15	100	0.0	100
Pr10	1.08	1.08	1.11	1.25	0.0708	100	0.0	88
Pr11	99.2	99.2	102.0	107.0	4.07	100	0.0	52
Pr12	2.92	2.95	3.0	4.21	0.254	100	0.0	16
Pr13	26900.0	26900.0	26900.0	26900.0	1.11E−11	100	0.0	100
Pr14	58500.0	66500.0	66000.0	73600.0	5140.0	100	0.0	0
Pr15	2990.0	2990.0	2990.0	2990.0	4.64E−13	100	0.0	100
Pr16	0.0322	0.0322	6.24	2.95	28.6	80	0.0128	80
Pr17	0.0127	0.0127	0.0127	0.0127	1.08E−5	100	0.0	100
Pr18	6060.0	6060.0	6060.0	6090.0	6.16	100	0.0	24
Pr19	1.67	1.67	1.67	1.67	1.2E−16	100	0.0	100
Pr20	264.0	264.0	264.0	264.0	0.0	100	0.0	100
Pr21	0.235	0.235	0.235	0.235	1.13E−16	100	0.0	100
Pr22	0.526	0.526	0.526	0.527	5.5E−4	100	0.0	36
Pr23	16.1	16.1	16.1	16.1	4.17E−15	100	0.0	100
Pr24	2.54	2.54	2.54	2.54	5.99E−14	100	0.0	100
Pr25	1860.0	260.0	1930.0	260.0	2370.0	40	9.01E−5	0
Pr26	45.4	49.1	62.1	45.5	29.3	12	0.135	0
Pr27	524.0	524.0	524.0	524.0	4.72E−4	100	0.0	88
Pr28	14600.0	14600.0	14600.0	14600.0	9.28E−12	100	0.0	100
Pr29	2960000.0	2960000.0	2960000.0	2960000.0	1.43E−9	100	0.0	100
Pr30	2.66	4.54	5.53	26.7	4.74	92	0.00181	44
Pr31	0.0	1.74E−18	4.55E−16	8.41E−15	1.68E−15	100	0.0	100
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	3.71E−12	100	0.0	100
Pr33	2.64	2.64	2.64	2.64	1.41E−15	100	0.0	100

Table 37: Statistical results of the ϵMAgES algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	189.0	492.0	455.0	437.0	223.0	84	1.47E−4	20
Pr02	7050.0	7800.0	7740.0	7480.0	750.0	96	3710.0	96
Pr03	−143.0	76.3	−160.0	495.0	888.0	92	0.0161	0
Pr04	−0.388	−0.375	−0.55	−1.0	0.286	72	0.035	24
Pr05	−400.0	−398.0	−363.0	−100.0	75.5	100	0.0	36
Pr06	2.09	1.83	1.59	1.2	0.339	0	0.0707	0
Pr07	2.0	1.71	1.8	2.01	0.174	0	0.0198	0
Pr08	2.0	2.0	1.99	1.29	0.152	96	0.00458	64
Pr09	2.56	2.56	2.55	1.93	0.27	92	0.0115	92
Pr10	1.08	1.08	1.08	1.25	0.0347	100	0.0	92
Pr11	107.0	99.2	106.0	115.0	6.88	0	0.022	0
Pr12	2.92	3.64	3.65	4.69	0.587	100	0.0	20
Pr13	26900.0	26900.0	26900.0	26900.0	1.11E−11	100	0.0	100
Pr14	53600.0	58500.0	58100.0	58500.0	1350.0	100	0.0	0
Pr15	2990.0	2990.0	2990.0	2990.0	4.64E−13	100	0.0	100
Pr16	0.0322	0.0322	0.0322	0.0322	2.78E−17	100	0.0	100
Pr17	0.0127	0.0127	0.0127	0.0137	2.16E−4	100	0.0	96
Pr18	6060.0	6410.0	7380.0	11900.0	1930.0	100	0.0	16
Pr19	1.67	1.67	1.69	1.85	0.0395	100	0.0	44
Pr20	264.0	264.0	265.0	274.0	2.88	100	0.0	88
Pr21	0.235	0.235	0.235	0.235	1.13E−16	100	0.0	100
Pr22	0.526	0.546	0.616	1.12	0.198	76	1.31	20
Pr23	16.1	16.1	16.1	16.1	1.78E−14	100	0.0	100
Pr24	2.54	2.54	2.54	2.55	8.81E−4	100	0.0	96
Pr25	1620.0	2260.0	2350.0	634.0	1410.0	88	1.84E−5	0
Pr26	82.3	21.3	33.8	7.26	36.1	8	0.377	0
Pr27	524.0	531.0	530.0	531.0	2.03	100	0.0	72
Pr28	14600.0	14600.0	14600.0	14600.0	9.28E−12	100	0.0	100
Pr29	2960000.0	2960000.0	2960000.0	2960000.0	1.43E−9	100	0.0	100
Pr30	2.66	3.11	2.21	0.0434	1.22	68	1040000.0	32
Pr31	0.0	0.0	0.0	0.0	0.0	100	0.0	100
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	3.56E−12	100	0.0	100
Pr33	2.64	2.65	2.65	2.68	0.0126	100	0.0	44

Table 38: Statistical results of the iLSHADE$_\epsilon$ algorithm on the considered COPs

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr01	190.0	194.0	206.0	229.0	19.3	28	0.0136	4
Pr02	7050.0	7050.0	7050.0	7050.0	2.13E−11	100	0.0	100
Pr03	−4530.0	−143.0	−818.0	534.0	1910.0	100	0.0	20

(Continued)

Table 38 (continued)

ID	Best	Median	Mean	Worst	STD	FR	MV	SR
Pr04	−0.375	−0.375	−0.375	−0.373	4.61E−4	100	0.0	0
Pr05	−400.0	−0.00806	−117.0	15.7	179.0	100	0.0	44
Pr06	1.72	1.24	1.27	1.11	0.163	0	3.94	0
Pr07	1.75	1.73	1.66	1.67	0.1	0	2.99	0
Pr08	2.0	2.0	2.0	2.0	0.0	100	0.0	100
Pr09	2.56	2.56	2.56	2.56	1.46E−7	100	0.0	100
Pr10	1.08	1.25	1.22	1.25	0.0648	100	0.0	88
Pr11	99.2	107.0	106.0	132.0	7.39	96	0.04	44
Pr12	2.92	2.92	2.92	2.92	8.14E−7	100	0.0	100
Pr13	26900.0	26900.0	26900.0	26900.0	1.11E−11	100	0.0	100
Pr14	53600.0	59200.0	59100.0	63600.0	1850.0	100	0.0	0
Pr15	2990.0	2990.0	2990.0	2990.0	4.64E−13	100	0.0	100
Pr16	0.0322	0.0323	0.382	2.95	0.967	88	0.116	72
Pr17	0.0127	0.0127	0.013	0.0178	0.00106	100	0.0	88
Pr18	6060.0	6110.0	8480.0	14900.0	3140.0	100	0.0	0
Pr19	1.67	1.67	1.67	1.67	7.59E−7	100	0.0	100
Pr20	264.0	264.0	264.0	264.0	0.0199	100	0.0	92
Pr21	0.235	0.235	0.235	0.235	1.13E−16	100	0.0	100
Pr22	0.526	0.526	0.527	0.531	0.00169	100	0.0	28
Pr23	16.1	16.1	16.1	16.1	8.62E−8	100	0.0	100
Pr24	2.54	2.54	2.54	2.55	0.00199	100	0.0	92
Pr25	1660.0	2090.0	1760.0	612.0	1280.0	76	2.54E−4	0
Pr26	35.4	36.3	36.3	37.3	0.659	100	0.0	24
Pr27	524.0	525.0	525.0	525.0	0.0714	100	0.0	0
Pr28	14600.0	14600.0	14600.0	14600.0	9.28E−12	100	0.0	100
RCy29	2960000.0	2960000.0	2970000.0	2970000.0	657.0	100	0.0	96
Pr30	6.9	2.87	6.26	20.5	6.32	20	0.795	0
Pr31	0.0	5.41E−18	5.56E−17	3.91E−16	1.17E−16	100	0.0	100
Pr32	−30700.0	−30700.0	−30700.0	−30700.0	3.64E−12	100	0.0	100
Pr33	2.64	2.64	2.64	2.64	8.11E−16	100	0.0	100

To assess and compare the performance of different optimizers on the set of 33 constrained optimization problems, we employ a comprehensive Evaluation Metric (EM), which will be calculated using Eq. (21). This equation incorporates a weighted sum of various key metrics, namely the best value attained throughout all optimization runs, the feasibility rate, the mean constraint violation, and the success rate. To emphasize the significance of each metric in the computation of EM, they are multiplied by specific coefficients: c_1, c_2, c_3, and c_4, reflecting their relative importance. By employing this multifaceted evaluation approach, we aim to gain a holistic understanding of the optimizers' effectiveness in addressing the considered constrained optimization problems.

$$\mathrm{EM} = c_1 \times \mathrm{Best} + c_2 \times \mathrm{FR} + c_3 \times \mathrm{MV} + c_4 \times \mathrm{SR} \tag{21}$$

To evaluate the significance of the obtained EM values and determine whether there exists a notable difference in performance among the algorithms examined in this comparative study, we subject the EM values to the Friedman test at a significance level of 0.05. The resulting p-value is $7.42 \cdot 10^{-11}$, indicating statistical evidence of performance discrepancies among the considered algorithms. This finding signifies that at least one algorithm exhibits a distinct level of performance compared to the others, warranting further investigation and analysis to identify the specific variations in their optimization capabilities.

To conduct a more detailed analysis of the performance differences among the optimizers, we further apply the post-hoc Dunn's test to the EM values. The resulting mean ranks, as presented in Table 39, provide valuable insights into the relative performance of each algorithm. Notably, the BHJO algorithm achieved the highest rank, indicating that it exhibits the best overall performance compared to the other optimizers under consideration. To delve deeper into the statistical significance of these performance disparities, we present the corresponding p-values in Table 40, enabling a comprehensive investigation and further exploration of the observed variations.

Table 39: Ranking of the optimizers considered for the comparative study

Optimizer	BHJO	BWO	HBA	IUDE	ϵMAgES	iLSHADE$_\epsilon$
Mean rank	2.12	2.76	3.42	3.74	5.33	3.62
Ranking	1	2	3	5	6	4

Table 40: Results of Dunn's test using the metric computed by Eq. (21)

	BHJO	BWO	HBA	IUDE	ϵMAgES	iLSHADE$_\epsilon$
BHJO	–	1.00E+00	7.00E−02	**1.00E−02**	**4.61E−11**	**2.00E−02**
BWO	–	–	1.00E+00	4.90E−01	**3.36E−07**	9.10E−01
HBA	–	–	–	1.00E+00	**5.10E−04**	1.00E+00
IUDE	–	–	–	–	**1.00E−02**	–
εMAgES	–	–	–	–	–	–
iLSHADE$_\epsilon$	–	–	–	1.00E+00	**0.00E+00**	–

5 Conclusion and Future Directions

In conclusion, this research paper presented a hybrid algorithm, BHJO, which combines the Beluga Whale Optimization, Honey Badger Algorithm, and Jellyfish Search optimizer. Through a meticulous analysis of the exploration and exploitation capabilities of these metaheuristics, their strengths, and weaknesses were identified and leveraged in the development of the BHJO optimizer. In other words, after a thorough evaluation of the three metaheuristics foundational to the BHJO algorithm, we have identified several strengths. The BWO and the HBA both show promising results in terms of exploitation and maintaining stability during exploration phases. However, they could benefit from an improved balance between these two aspects. On the other hand, the JS excels in exploration, offering a commendable balance between exploration and exploitation, although it does fall short in its exploitation capabilities. Collectively, these attributes contribute to the overall effectiveness of the BHJO algorithm, making each component valuable in its own right despite some areas needing enhancement, and this is shown via the thorough experimental study. Additionally, the incorporation of Opposition-Based Learning further enhanced the algorithm's performance and solution quality. Extensive evaluations on both unconstrained and constrained optimization problems demonstrated the efficacy and versatility of the BHJO algorithm across diverse domains. Comparative analyses against renowned algorithms confirmed the algorithm's competitiveness. Furthermore, the statistical analysis through the Friedman post hoc Dunn's test revealed significant performance differences, highlighting the superiority of the BHJO optimizer in solving complex optimization problems. Overall, this research introduces a promising hybrid algorithm with immense potential for addressing challenging optimization tasks and contributes to the advancement of metaheuristic algorithms. In the interest of a balanced discussion, it is important to acknowledge some limitations of the BHJO algorithm. Firstly, the algorithm includes multiple parameters that require meticulous adjustment to reach optimal performance, a process that is both time-consuming and dependent on extensive experimental or simulation efforts. Additionally, the BHJO algorithm is computationally demanding, largely due to its integration of several heuristic strategies, each of which independently consumes significant resources. Moreover, there is a risk of the algorithm becoming trapped in local optima, as there are no definitive guarantees regarding the effectiveness of its swarming process. By addressing these concerns, the effectiveness of the BHJO algorithm can be significantly enhanced, leading to improved performance and reliability.

In the future, several directions can be pursued to further enhance the BHJO algorithm. For example, exploring the integration of machine learning techniques, such as deep learning or reinforcement learning, into the BHJO algorithm can provide opportunities for improved learning capabilities and adaptive behavior. This future direction aims to advance the algorithm's versatility, efficiency, and applicability in solving a wider range of optimization problems.

Acknowledgement: The authors present their appreciation to King Saud University for funding the publication of this research through Researchers Supporting Program (RSPD2024R809), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: The research is funded by the Researchers Supporting Program at King Saud University (RSPD2024R809).

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Farouq Zitouni; data collection: Farouq Zitouni, Fatima Zohra Khechiba, Khadidja Kherchouche; analysis and interpretation of results: Farouq Zitouni, Fatima Zohra Khechiba, Khadidja Kherchouche; draft manuscript preparation: Farouq Zitouni, Saad Harous, Abdulaziz S. Almazyad, Ali Wagdy Mohamed, Guojiang Xiong. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in this published article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Mohamed A, Oliva D, Suganthan PN. Handbook of nature-inspired optimization algorithms: the state of the art. In: Studies in systems, decision and control, Cham: Springer; 2022. vol. 212.
2. Ye H, Gu Y, Zhang X, Wang S, Tan F, Zhang J, et al. Fast optimization for Betatron radiation from laser wakefield acceleration based on Bayesian optimization. Results Phys. 2022;43:106116. doi:10.1016/j.rinp.2022.106116.
3. Abaluck J, Gruber J. When less is more: improving choices in health insurance markets. Rev Econ Stud. 2023;90(3):1011–40. doi:10.1093/restud/rdac050.
4. Weinberg SJ, Sanches F, Ide T, Kamiya K, Correll R. Supply chain logistics with quantum and classical annealing algorithms. Sci Rep. 2023;13(1):4770. doi:10.1038/s41598-023-31765-8.
5. Nicoletti A, Martino M, Karimi A. A data-driven approach to power converter control via convex optimization. In: 1st IEEE Conference on Control Technology and Applications, 2017; Kohala Coast, Hawaii; p. 1466–71.
6. Krentel MW. The complexity of optimization problems. In: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, 1986; New York, NY, USA: Association for Computing Machinery; p. 69–76. doi:10.1145/12130.12138.
7. Okwu MO, Tartibu LK. Metaheuristic optimization: nature-inspired algorithms swarm and computational intelligence, theory and applications. In: Studies in computational intelligence, Cham: Springer; 2021. vol. 927.
8. Kaul S, Kumar Y. Nature-inspired metaheuristic algorithms for constraint handling: challenges, issues, and research perspective. In: Constraint handling in metaheuristics and applications. Singapore: Springer; 2021. p. 55–80.
9. Chou X, Gambardella LM, Montemanni R. A metaheuristic algorithm for the probabilistic orienteering problem. In: Proceedings of the 2019 2nd International Conference on Machine Learning and Machine Intelligence, 2019; New York, NY, USA: Association for Computing Machinery; p. 30–34. doi:10.1145/3366750.3366761.
10. Tsai CW, Chiang MC. Handbook of metaheuristic algorithms: from fundamental theories to advanced applications. Elsevier; 2023. p. 71–76. https://books.google.dz/books?id=P6umEAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. [Accessed 2024].
11. Keller AA. Multi-objective optimization in theory and practice II: Metaheuristic algorithms. Bentham Science Publishers; 2019. p. 54–81. https://books.google.dz/books?id=3uyRDwAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. [Accessed 2024].
12. Carbas S, Toktas A, Ustun D. Introduction and overview: nature-inspired metaheuristic algorithms for engineering optimization applications. In: Nature-inspired metaheuristic algorithms for engineering optimization applications, Singapore: Springer Singapore; 2021. p. 1–9. doi:10.1007/978-981-33-6773-9_1.
13. Holland JH. Genetic algorithms. Sci Am. 1992;267(1):66–73. doi:10.1038/scientificamerican0792-66.
14. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, 1995; Perth, WA, Australia: IEEE; vol. 4, p. 1942–8.
15. Bertsimas D, Tsitsiklis J. Simulated annealing. Stat Sci. 1993;8(1):10–5.
16. Glover F. Tabu search: a tutorial. Interfaces. 1990;20(4):74–94.
17. Greiner D, Periaux J, Quagliarella D, Magalhaes-Mendes J, Galván B. Evolutionary algorithms and metaheuristics: applications in engineering design and optimization. Math Probl Eng. 2018;2018:1–4.
18. Xhafa F, Abraham A. Metaheuristics for scheduling in industrial and manufacturing applications. Springer Berlin, Heidelberg: Springer; 2008. vol. 128.
19. Jain DK, Tyagi SKS, Neelakandan S, Prakash M, Natrayan L. Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment. IEEE Trans Ind Inform. 2021;18(7):4884–92.

20. Dhaenens C, Jourdan L. Metaheuristics for data mining: survey and opportunities for big data. Ann Oper Res. 2022;314(1):117–40.
21. Monteiro ACB, França RP, Estrela VV, Razmjooy N, Iano Y, Negrete PDM. Metaheuristics applied to blood image analysis. In: Metaheuristics and optimization in computer and electrical engineering. Springer; 2020. p. 117–135.
22. Zhong C, Li G, Meng Z. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm. Knowl Based Syst. 2022;251:109215.
23. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. Math Comput Simul. 2022;192:84–110.
24. Chou JS, Truong DN. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. Appl Math Comput. 2021;389:125535.
25. Mohammadi-Balani A, Nayeri MD, Azar A, Taghizadeh-Yazdi M. Golden eagle optimizer: a nature-inspired metaheuristic algorithm. Comput Ind Eng. 2021;152:107050.
26. Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S. Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Softw. 2022;174:103282.
27. Zhao W, Wang L, Zhang Z. Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput Appl. 2020;32:9383–425.
28. Zhao S, Zhang T, Ma S, Chen M. Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. Eng Appl Artif Intell. 2022;114:105075.
29. Zitouni F, Harous S, Belkeram A, Hammou LEB. The archerfish hunting optimizer: a novel metaheuristic algorithm for global optimization. Arab J Sci Eng. 2022;47(2):2513–53.
30. Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng. 2021;158:107408.
31. Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int J Intell Syst. 2021;36(10):5887–958.
32. Jafari M, Salajegheh E, Salajegheh J. Elephant clan optimization: a nature-inspired metaheuristic algorithm for the optimal design of structures. Appl Soft Comput. 2021;113:107892.
33. Akbari MA, Zare M, Azizipanah-Abarghooee R, Mirjalili S, Deriche M. The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. Sci Rep. 2022;12(1):10953.
34. Ghasemi-Marzbali A. A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. Soft Comput. 2020;24(17):13003–35.
35. Mohamed AW, Hadi AA, Mohamed AK. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. Int J Mach Learn Cybern. 2020;11(7):1501–29.
36. Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M. Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowl Based Syst. 2023;262:110248.
37. Kumar N, Singh N, Vidyarthi DP. Artificial lizard search optimization (ALSO): a novel nature-inspired meta-heuristic algorithm. Soft Comput. 2021;25(8):6179–201.
38. Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R. Red deer algorithm (RDA): a new nature-inspired meta-heuristic. Soft Comput. 2020;24:14637–65.
39. Oyelade ON, Ezugwu AES, Mohamed TI, Abualigah L. Ebola optimization search algorithm: a new nature-inspired metaheuristic optimization algorithm. IEEE Access. 2022;10:16150–77.
40. Zitouni F, Harous S, Maamri R. The solar system algorithm: a novel metaheuristic method for global optimization. IEEE Access. 2020;9:4542–65.

41. Trojovský P, Dehghani M, Hanuš P. Siberian tiger optimization: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems. IEEE Access. 2022;10:132396–431.
42. Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl Based Syst. 2023;259:110011.
43. Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell. 2022;114:105082.
44. Rabie AH, Saleh AI, Mansour NA. Red piranha optimization (RPO): a natural inspired meta-heuristic algorithm for solving complex optimization problems. J Ambient Intell Humaniz Comput. 2023;14(6): 7621–48.
45. Harifi S, Mohammadzadeh J, Khalilian M, Ebrahimnejad S. Giza pyramids construction: an ancient-inspired metaheuristic algorithm for optimization. Evol Intell. 2021;14:1743–61.
46. Zhao S, Zhang T, Ma S, Wang M. Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl Intell. 2023;53(10):11833–60.
47. Eslami N, Yazdani S, Mirzaei M, Hadavandi E. Aphid–ant mutualism: a novel nature-inspired metaheuristic algorithm for solving optimization problems. Math Comput Simul. 2022;201:362–95.
48. Braik M, Hammouri A, Atwan J, Al-Betar MA, Awadallah MA. White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl Based Syst. 2022;243:108457.
49. Ghasemi M, Akbari MA, Jun C, Bateni SM, Zare M, Zahedi A, et al. Circulatory system based optimization (CSBO): an expert multilevel biologically inspired meta-heuristic algorithm. Eng Appl Comput Fluid Mech. 2022;16(1):1483–525.
50. Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH. Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst Appl. 2022;191:116158.
51. Yadav D. Blood coagulation algorithm: a novel bio-inspired meta-heuristic algorithm for global optimization. Mathematics. 2021;9(23):3011.
52. Zaeimi M, Ghoddosian A. Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization. Soft Comput. 2020;24:12027–66.
53. Rabie AH, Mansour NA, Saleh AI. Leopard seal optimization (LSO): A natural inspired meta-heuristic algorithm. Commun Nonlinear Sci Numer Simul. 2023;125:107338.
54. Dehghani M, Trojovský P. Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems. Front Mech Eng. 2023;8:1126450.
55. Dehghani M, Trojovský P, Malik OP. Green anaconda optimization: a new bio-inspired metaheuristic algorithm for solving optimization problems. Biomimetics. 2023;8(1):121.
56. Trojovský P, Dehghani M. Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems. Biomimetics. 2023;8(2):149.
57. Zitouni F, Harous S, Maamri R. A novel quantum firefly algorithm for global optimization. Arab J Sci Eng. 2021;46(9):8741–59.
58. Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput. 1997;1(1):67–82.
59. Viswanathan G, Afanasyev V, Buldyrev SV, Havlin S, Da Luz M, Raposo E, et al. Lévy flights in random searches. Phys A: Stat Mech Appl. 2000;282(1–2):1–12.
60. Adelberger EG, Heckel BR, Nelson AE. Tests of the gravitational inverse-square law. Annu Rev Nucl Part Sci. 2003;53(1):77–121.
61. Xiang T, Liao X, Wong Kw. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Appl Math Comput. 2007;190(2):1637–45. doi:10.1016/j.amc.2007.02.103.
62. Chou JS, Ngo NT. Modified firefly algorithm for multidimensional optimization in structural design problems. Struct Multidiscipl Optim. 2017;55:2013–28. doi:10.1007/s00158-016-1624-x.

63. Kaveh A, Mahdipour Moghanni R, Javadi S. Optimum design of large steel skeletal structures using chaotic firefly optimization algorithm based on the Gaussian map. Struct Multidiscipl Optim. 2019;60:879–94. doi:10.1007/s00158-019-02263-1.
64. Gandomi AH, Yang XS, Talatahari S, Alavi AH. Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul. 2013;18(1):89–98. doi:10.1016/j.cnsns.2012.06.009.
65. Kıran MS, Fındık O. A directed artificial bee colony algorithm. Appl Soft Comput. 2015;26:454–62. doi:10.1016/j.asoc.2014.10.020.
66. Fossette S, Gleiss AC, Chalumeau J, Bastian T, Armstrong CD, Vandenabeele S, et al. Current-oriented swimming by jellyfish and its role in bloom maintenance. Curr Biol. 2015;25(3):342–7. doi:10.1016/j.cub.2014.11.050.
67. Zavodnik D. Spatial aggregations of the swarming jellyfish Pelagia noctiluca (Scyphozoa). Mar Biol. 1987;94(2):265–9. doi:10.1007/BF00392939.
68. Gaidhane PJ, Nigam MJ. A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. J Comput Sci. 2018;27:284–302. doi:10.1016/j.jocs.2018.06.008.
69. Talbi EG. A taxonomy of hybrid metaheuristics. J Heuristics. 2002;8:541–64. doi:10.1023/A:1016540724870.
70. Liu Y, Li X. A hybrid mobile robot path planning scheme based on modified gray wolf optimization and situation assessment. J Robot. 2022;2022:1–9. doi:10.1155/2022/4167170.
71. Xiao Y, Guo Y, Cui H, Wang Y, Li J, Zhang Y. IHAOAVOA: an improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems. Math Biosci Eng. 2022;19(11):10963–1017.
72. Xiao Y, Sun X, Guo Y, Li S, Zhang Y, Wang Y. An improved gorilla troops optimizer based on lens opposition-based learning and adaptive β-hill climbing for global optimization. Comput Model Eng Sci. 2022;131(2):815–50. doi:10.32604/cmes.2022.019198.
73. Tizhoosh H. Opposition-based learning: a new scheme for machine intelligence. In: Paper presented at the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2005;Vienna, Austria.
74. Digalakis JG, Margaritis KG. On benchmarking functions for genetic algorithms. Int J Comput Math. 2001;77(4):481–506.
75. Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput. 1999;3(2):82–102.
76. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95:51–67.
77. Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst. 2015;89:228–49.
78. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. Future Gener Comput Syst. 2019;97:849–72.
79. Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol Comput. 2020; 56:100693.
80. Trivedi A, Srinivasan D, Biswas N. An improved unified differential evolution algorithm for constrained optimization problems. In: Proceedings of 2018 IEEE Congress on Evolutionary Computation, 2018; Rio de Janeiro, Brazil: IEEE. doi:10.1109/CEC.2017.7969446.

81. Hellwig M, Beyer HG. A matrix adaptation evolution strategy for constrained real-parameter optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018; Rio de Janeiro, Brazil: IEEE; p. 1–8. doi:10.1109/CEC.2018.8477950.
82. Fan Z, Fang Y, Li W, Yuan Y, Wang Z, Bian X. LSHADE44 with an improved epsilon constraint-handling method for solving constrained single-objective optimization problems. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018; Rio de Janeiro, Brazil: IEEE; p. 1–8. doi:10.1109/CEC.2018.8477943.