

Hijrah Search Optimizer: A Spiritual Migration-Inspired Metaheuristic Algorithm for Solving Engineering Design Problems

Moh Nur Sholeh

`mns@live.undip.ac.id`

Diponegoro University

Linda Karlina

National Taiwan University of Science and Technology

Research Article

Keywords: Hijrah search optimizer, Metaheuristic algorithm, Optimization, Opportunity control mechanism, Engineering design problems

Posted Date: September 18th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7334045/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Hijrah Search Optimizer: A Spiritual Migration-Inspired Metaheuristic Algorithm for Solving Engineering Design Problems

Moh Nur Sholeh^{1*}, Linda Karlina²

¹Department of Civil and Planning, Diponegoro University, Semarang, Indonesia

²Graduate Institute of Applied Science and Technology, National Taiwan University of Science and Technology, Taipei, Taiwan

***Corresponding author: Moh Nur Sholeh, Ph.D.**

Assistant Professor, Department of Civil and Planning, Diponegoro University, Semarang, Indonesia

Address : Jl. Prof. Soedarto, SH., Tembalang, Semarang 50275, Indonesia

E-mail address : mns@live.undip.ac.id

Orchid ID : 0000-0002-2795-9540

Linda Karlina

Ph.D. Student, Graduate Institute of Applied Science and Technology, National Taiwan University of Science and Technology, Taipei, Taiwan

Address : No. 43, Sec. 4, Keelung Rd., Taipei, 106, Taiwan (ROC)

E-mail address : D11122802@mail.ntust.edu.tw; lindakarlinans@gmail.com

Orchid ID : NA

Hijrah Search Optimizer: A Spiritual Migration-Inspired Metaheuristic Algorithm for Solving Engineering Design Problems

Abstract

This study develops the Hijrah Search Optimizer (HSO), a novel metaheuristic algorithm inspired by the Islamic concept of *hijrah*, the spiritual migration from unfavorable conditions (Makkah) toward a better state (Madinah). The simulation of migration behavior includes strategies such as moving openly, moving covertly (following a guide and taking an unusual route), and opportunity control mechanisms that switch between these movements, culminating in convergence upon arrival at a better state. HSO is evaluated using a comprehensive set of mathematical benchmark functions and applied to a series of engineering design problems. These benchmark functions are used to validate HSO's performance, which is compared with several well-known metaheuristic algorithms. The results demonstrate that HSO outperforms existing algorithms in solving mathematical benchmark functions including thirty standard functions and CEC-2020 test suite. Additionally, HSO is successfully applied to engineering design problems, including welded beam and truss structure designs, where it not only achieves superior performance but also requires fewer objective function evaluations. These findings indicate that HSO is a promising metaheuristic algorithm for solving complex optimization problems.

Keywords: Hijrah search optimizer; Metaheuristic algorithm; Optimization; Opportunity control mechanism; Engineering design problems.

1. Introduction

The pursuit of optimal solutions has attracted significant interest from the research community, owing to its vital role in addressing complex, real-world optimization problems. Optimization is fundamentally an iterative improvement process that seeks the best configuration within a given search space. These spaces are often characterized by challenging properties such as multimodality, non-convexity, non-differentiability, and the presence of various constraints [1]. Depending on the nature of their variables, binary, discrete, continuous, permutation-based, or structured, optimization problems require customized strategies to navigate effectively. Such challenges are prevalent in a wide range of domains, including engineering [2, 3], scheduling [4, 5], computer vision [6], feature selection [7, 8], image processing [9], industrial system modeling [10], and among others. To address these complex tasks, researchers have increasingly turned to metaheuristic optimization algorithms.

Metaheuristic algorithms have emerged as powerful, general-purpose methods capable of solving high-dimensional and nonlinear problems. Typically, these algorithms begin with a population of randomly generated solutions, which are iteratively refined using intelligent, parameterized operators. These operators facilitate a balance between exploration, searching new areas of the solution space, and exploitation, refining known good solutions [11, 12]. Notable advantages of MH algorithms include their derivative-free nature, simplicity, adaptability, and strong global search capabilities, which allow them to avoid being trapped in local optima [13]. Inspired by processes in nature, physics, society, and behavior, MH algorithms are often referred to as nature-inspired algorithms. These can be broadly categorized into swarm-based, evolutionary-based, physics- or chemistry-based, and socially- or human-inspired methods [14, 15], which are discussed in detail in **Section 2**.

Despite their shared structure and guiding principles, the effectiveness of MH algorithms can vary significantly depending on the problem landscape, a reality supported by the No Free Lunch (NFL) theorem [16, 17]. Therefore, continuous innovation remains essential in developing new algorithms that better balance exploration and exploitation for specific optimization tasks. Many existing metaheuristic (MH) algorithms have drawn inspiration from human group behaviors, such as teamwork in sports [18], decision-making in social communities [19], and collaborative problem-solving in organizations [20], mimicking collective intelligence and coordination strategies. While these behavior-inspired approaches have demonstrated success, there remain opportunities to explore new sources of inspiration that capture different aspects of the optimization process.

This study proposes the Hijrah Search Optimizer (HSO), a novel metaheuristic algorithm designed to address optimization problems. Inspired by the Islamic concept of *hijrah*, the spiritual migration from unfavorable conditions (Makkah) to a better state (Madinah), HSO is benchmarked against recent and widely used metaheuristic algorithms. However, according to the No Free Lunch theorem [xx], no algorithm can universally outperform others across all problems. Therefore, it is essential to carefully analyze and adapt metaheuristic algorithms to specific problem domains to enhance their efficiency and effectiveness.

The key contributions of this study are as follows:

- A novel metaheuristic algorithm, Hijrah Search Optimizer (HSO), is developed, inspired by the migration behavior of agents moving from unfavorable conditions to a better state.
- An opportunity control mechanism is introduced to enhance efficiency by dynamically switching between open movement during exploration and covert movement during exploitation.
- HSO's performance is validated using thirty benchmark functions and the CEC-2020 test suite.
- Comprehensive statistical analyses, including the Wilcoxon signed-rank test and Friedman rank test, are performed to compare HSO with other leading algorithms.
- HSO is applied to real-world engineering design problems, such as welded beam and truss structure designs, and is benchmarked against reference algorithms.
- Experimental results confirm that HSO achieves superior performance, outperforming competing algorithms in both benchmark tests and engineering applications.

The rest of this paper is organized as follows: **Section 2** provides a review of related metaheuristic algorithms. **Section 3** introduces the HSO algorithm, including its inspiration and mathematical formulation. **Section 4** presents experimental results and a comparative performance analysis. **Section 5** concludes the paper and discusses future research directions.

2. Related Metaheuristic Algorithms

Metaheuristic algorithms have emerged as powerful tools for solving complex optimization problems across diverse domains. These algorithms are generally inspired by natural, biological, physical, or social processes and are designed to balance exploration (global search) and exploitation (local search) in the solution space. Over the years, various metaheuristics have been developed, ranging from evolutionary-based and swarm-based methods to those inspired by human behavior and cultural practices. Metaheuristic algorithms are generally classified into four major categories: evolutionary algorithms (EAs), swarm

intelligence (SI) algorithms, law-based algorithms, and human-inspired algorithms [21]. Evolutionary algorithms are modeled on biological evolution and employ operators such as crossover, mutation, and selection. Swarm intelligence algorithms are inspired by collective animal behaviors, such as foraging, flocking, or territorial interactions, to navigate the search space efficiently. Law-based algorithms are founded on mathematical, physical, or chemical principles to guide the search process. Human-inspired algorithms draw on social behaviors, decision-making processes, and collaborative dynamics observed in human societies. Representative algorithms from each of these categories are summarized in **Table 1**.

Table 1. Categories and examples of metaheuristic algorithms

Category	Algorithms	Authors	Year	Inspiration
Evolutionary algorithms	Genetic Algorithm (GA)	Holland [22]	1992	Natural selection and genetics
	Differential Evolution (DE)	Price [23]	1996	Population-based differential mutation and crossover
	Biogeography-Based Optimization (BBO)	Simon [24]	2008	Migration and distribution of species
	Pine Cone Optimization Algorithm (PCOA)	Anaraki & Farzin [25]	2024	Pine tree reproduction
Swarm intelligence algorithms	Particle Swarm Optimization (PSO)	Kennedy & Eberhart [26]	1995	Bird flocking and fish schooling
	Grey Wolf Optimizer (GWO)	Mirjalili et al. [27]	2014	Hunting strategy of grey wolves
	Salp Swarm Algorithm (SSA)	Mirjalili et al. [28]	2017	Swarming behavior of salps
	Tuna Swarm Optimization (TSA)	Xie et al. [29]	2021	Foraging behavior of tuna swarm
	Spider Wasp Optimizer (SWO)	Abdel-Basset et al. [30]	2023	Behavior of the female spider wasps
	Birds-of-Paradise Search (BPS)	Sholeh & Linda [31]	2025	Behavior of birds-of-paradise in tropical rainforest
Law-based algorithms	Gravitational Search Algorithm (GSA)	Rashedi et al. [32]	2009	Laws of gravity and mass
	Electromagnetic Field Optimization (EFO)	Abedinpourshotorban et al. [33]	2016	Electromagnet in different polarities
	Archimedes Optimization Algorithm (AOA)	Hashim et al. [34]	2021	Archimedes' Principle
	Prism Refraction Search (PRS)	Kundu et al. [35]	2024	Light refraction through a prism
Human-inspired algorithms	Teaching-Learning-Based Optimization (TLBO)	Rao et al. [36]	2011	Teaching-learning process
	Human Mental Search (HMS)	Mousavirad & Ebrahimpour-Komleh [37]	2017	Online auction bidding strategies
	Doctor and Patient Optimizer (DPO)	Dehghani et al. [38]	2020	Process of treating patients by a physician
	Political Optimizer (PO)	Askari et al. [39]	2020	Multi-phased process of politics
	Driving Training-Based Optimization (DTBO)	Dehghani et al. [40]	2022	Driving lessons and instructor training
	Divine Religions Algorithm (DRA)	Mozhdehi et al. [41]	2025	Religious societies

3. Hijrah Search Optimizer

This section describes the inspiration behind, and the mathematical simulations involved in, the Hijrah Search Optimizer.

3.1. Inspiration

The Hijrah Search Optimizer (HSO) is inspired by the historical event of *hijrah*, the migration of the Prophet Muhammad and his companions from Makkah to Madinah. This migration was not a single event but a structured process involving different strategies and phases. Before the Prophet's own migration, some of his companions had already migrated in small groups and openly traveled to Madinah. Their migration represented an early phase of movement toward a better state, despite facing risks and challenges along the way.

The Prophet's own migration, however, was executed with a higher level of strategy and caution. It involved covert planning, including hiding in the Cave of Thawr for three days to avoid pursuers, following an unusual route guided by an experienced tracker, and moving primarily under the cover of night. This phase reflects adaptive behavior in response to threats, ensuring survival while maintaining progress toward the destination. Upon arrival in Madinah, the migrants formed cooperative bonds with the local inhabitants, known as the *Ansar* (helpers), establishing a unified and strengthened community. This phase symbolizes convergence and stability, where the search reaches its goal and consolidates gains.

The *hijrah* walk process, with its combination of open migration, covert movements, and final convergence, mirrors the key components required in an optimization algorithm: exploration, exploitation, and convergence. In the proposed HSO, the search process is modeled in three main phases:

- (a) **Open movement phase:** Inspired by the early companions' migration, this phase emphasizes exploration, allowing agents to traverse broad solution spaces in search of promising areas.
- (b) **Covert movement phase:** Reflecting the Prophet's strategic migration, this phase focuses on exploitation, where agents refine their search within narrowed regions, guided by opportunity control mechanisms to switch between paths.
- (c) **Convergence phase:** Echoing the settlement and cooperation in Madinah, this phase drives the search process toward stability and optimal solutions, ensuring that agents gather and consolidate around the best-found positions.

Thus, the HSO algorithm captures the essence of the *hijrah* journey, transforming it into a structured search mechanism for solving complex optimization problems. **Fig. 1** illustrates

the overall steps of the algorithm. In the next subsection, the migration-inspired behaviors are mathematically modeled and integrated into the HSO framework.

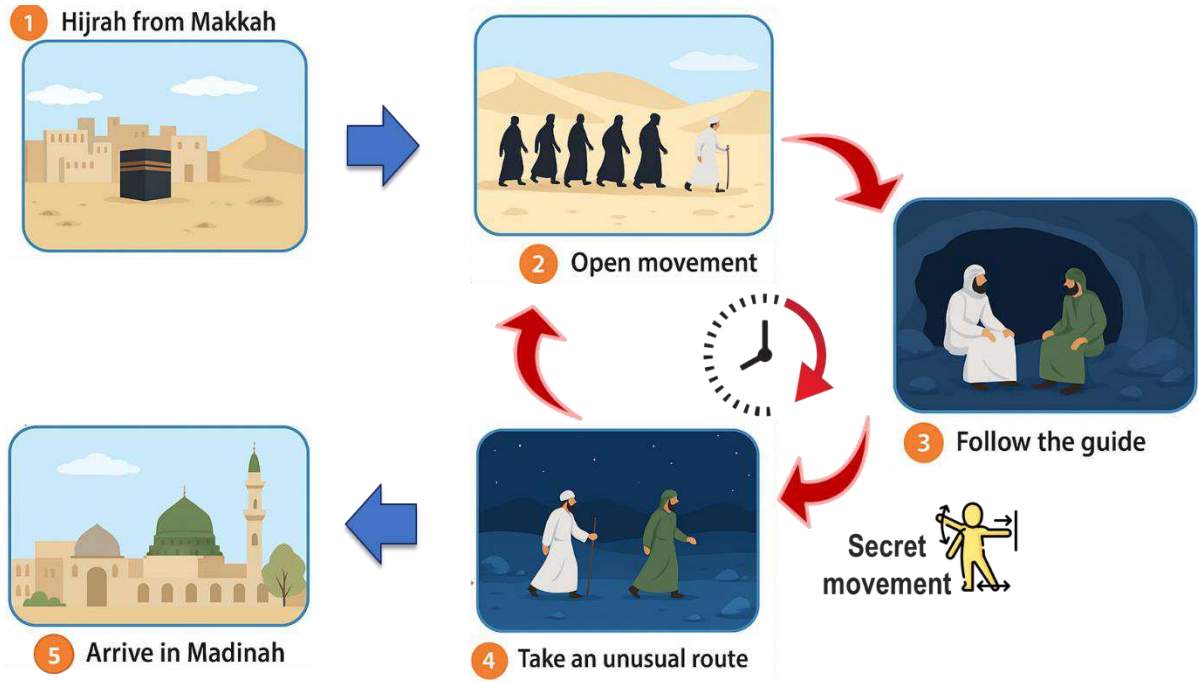


Fig. 1. Hijrah search behavior from Makkah to Madinah

3.2. Mathematical Model for HSO Algorithm

3.2.1. Initialization Process

The initialization of agent populations is commonly performed through random generation. However, this approach often suffers from limited diversity, which can slow convergence and increase the likelihood of entrapment in local optima. To overcome these limitations, the present study replaces conventional random initialization with a logistic chaotic map, aiming to enhance population diversity and mitigate the risk of premature convergence, as formulated in **Eq. (1)**:

$$A_{i+1} = \beta * A_i * (1 - A_i), 0 \leq A_i \leq 1 \quad (1)$$

where A_i represents the logistic chaotic position value of the i -th agent, with $A_0 \in (0,1)$, $A_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. The parameter β is fixed at 4.0 to ensure that the generated values remain within the interval $[0,1]$ [42]. After evaluating the fitness of the initial population, HSO determines the global best solution (A_{best}), which indicates the optimal agent position within the search space.

3.2.2. Open Movement

The open movement reflects the spirit of *hijrah*, where agents seek better opportunities by moving towards promising areas in the search space. This movement encourages agents to explore globally while avoiding stagnation. The direction of open movement is determined by the vector difference between the current agent and a randomly selected agent from the population, as illustrated in **Fig. 2**. **Eq. (2)** simulates the open movement mechanism.

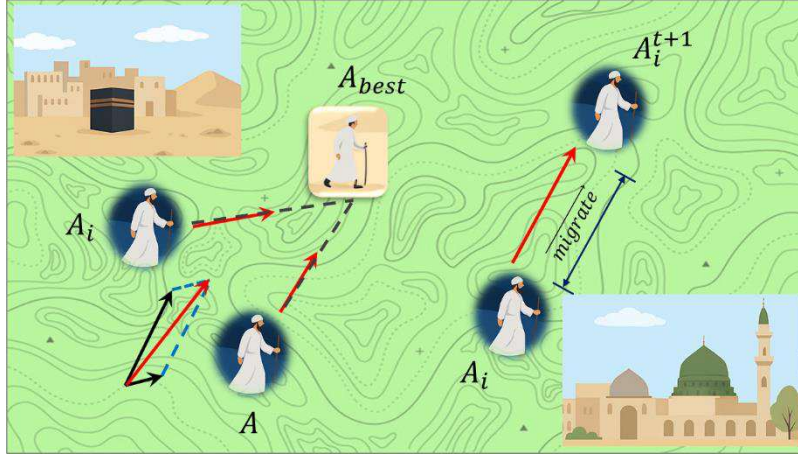


Fig. 2. Simulation of open movement from Makkah to Madinah

$$A_i^{t+1} = A_i^t + rand * \overrightarrow{A_{migrate}} \quad (2)$$

where A_i^{t+1} is the updated position of the agent, A_i^t represents the current position of the agent, *rand* is a random factor scaled according to the number of decision variables. The term $\overrightarrow{A_{migrate}}$ defines the directional movement of the agent toward more promising regions of the search space, as detailed in the following **Eq. (3)**:

$$A_{migrate} = r_v * A_{mean} - A_{best} \quad (3)$$

Here, A_{mean} is the average position of all agents in the population, and A_{best} refers the best-known position found so far. The variable r_v is a random coefficient that modulates trade-off during migration and is defined in **Eq. (4)**:

$$r_v = r_{[0,1]} * \frac{P}{T} \quad (4)$$

In this equation, $r_{[0,1]}$ is a random number uniformly distributed between 0 and 1, P denotes the number of migration points (which is set to 3 in this study), and T represents the migration period, set to 4 to correspond to the seasonal migration cycle (i.e., one year).

Substituting **Eq. (3)** and **Eq. (4)** into **Eq. (2)** results in the full migration update formula:

$$A_i^{t+1} = A_i^t + rand * \left(\left(r_{[0,1]} * \frac{P}{T} \right) * A_{mean} - A_{best} \right) \quad (5)$$

This formulation incorporates the concept of collective movement towards better regions of the search space, where agents are guided not only by the best-found solution but also by the average position of all agents, promoting both exploration and exploitation throughout the optimization process. The random coefficient r_v introduces variability into the migration dynamics, allowing for enhanced exploration in the early stages of the search and more focused exploitation in later stages.

3.2.3. Following a Guide During Secret Movement

In secret movement, all agents perform both passive (following a guide) and active (taking an unusual route) movements. Initially, when the opportunity to emigrate is more favorable, most agents exhibit follow a guide motion. Over time, they increasingly exhibit take an unusual route motion.

An agent (j) is randomly selected, and the vector from the desired agent (i) to the designated agent (j) is used to determine the movement action. The goal is for each agent to move in a better direction towards targets. The direction of action and the updated location of the agent are simulated by **Eqs. (6) and (7)**, respectively, with their movement illustrated in **Fig. 3**. This action is to effectively exploit the one-on-one search space.

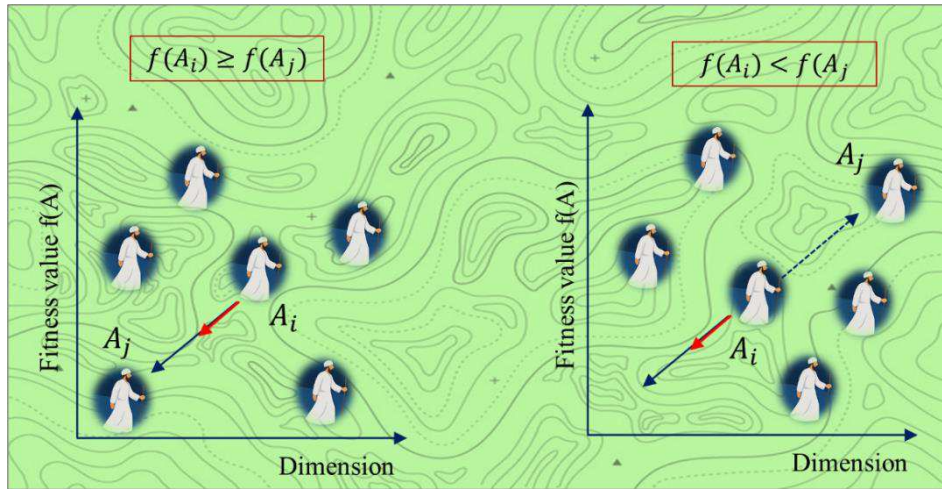


Fig. 3. Direction of agents following a guide during secret movement

The movement vector, $(\overrightarrow{move})$, is calculated as the difference between the new position, A_i^{t+1} , and the current position, A_i , as follow:

$$\overrightarrow{move} = A_i^{t+1} - A_i^t \quad (6)$$

The movement vector is then scaled by the random factor $rand$ and the direction vector $(\overrightarrow{dir})$, defined as:

$$\overrightarrow{move} = rand * \overrightarrow{dir} \quad (7)$$

The step vector is determined based on the fitness values of agent A_i and agent A_j , according to **Eq. (8)**.

$$\overrightarrow{dir} = \begin{cases} A_j^t - A_i^t & \text{if } f(A_i) \geq f(A_j) \\ A_i^t - A_j^t & \text{if } f(A_i) < f(A_j) \end{cases} \quad (8)$$

If the fitness of agent A_i is greater than or equal to the fitness of agent A_j , the action vector points from agent A_i to agent A_j . Conversely, if the fitness of agent A_i is lower than the fitness of agent A_j , the action vector points from agent A_j to agent A_i . Finally, the updated agent position, A_i^{t+1} , is calculated by adding the approach vector to the current position, as shown in **Eq. (9)**.

$$A_i^{t+1} = A_i^t + rand * \begin{cases} A_j^t - A_i^t & \text{if } f(A_i) \geq f(A_j) \\ A_i^t - A_j^t & \text{if } f(A_i) < f(A_j) \end{cases} \quad (9)$$

By following the guide mechanism, agents move towards promising areas based on fitness comparison, enhancing local exploitation and aiding optimal solution search during the *hijrah* walk.

3.2.4. Taking an Unusual Route During Secret Movement

In the context of the HSO algorithm, this phase is inspired by the strategic decision-making and actions undertaken during the Hijrah journey. Just as the early agents carefully navigated unfamiliar territories to ensure safety and discover new opportunities, the agents in HSO engage in taking an unusual route to intensify exploitation within promising areas of the search space. This movement simulates the behavior of a strategic salesperson, actively seeking the best prospects, adapting their route, and positioning themselves to secure optimal outcomes. It balances aggressive exploration with careful adjustment, reflecting the adaptability and foresight demonstrated during the Hijrah.

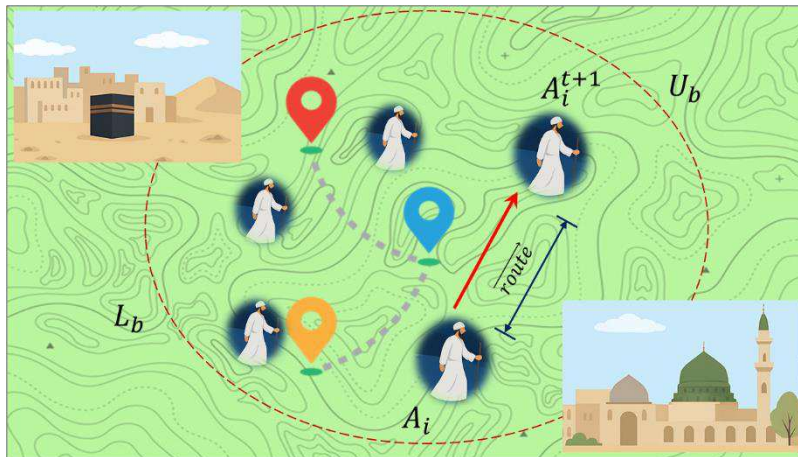


Fig. 4. Simulation of taking an unusual route from Makkah to Madinah

The taking an unusual route movement is mathematically modeled using a set of trigonometric functions that enable dynamic and directional adjustments:

$$\alpha = -1 + 2 * r_{[0,1]} \quad (10)$$

$$\rho = r_{[0,1]} * e^{r_{[0,1]}} \quad (11)$$

$$\theta_1 = \rho * \sin(\alpha) \quad (12)$$

$$\theta_2 = \rho * \cos(\alpha) \quad (13)$$

Here, α is a randomly distributed angle in the range $[-1,1]$, ρ is the adaptive radius controlling the movement scale, and e is the base of the natural logarithm. The variables θ_1 and θ_2 represent the directional displacement components based on the agent's movement behavior.

The magnitude of the movement, representing the agent's strategic selling behavior, is calculated as:

$$\overrightarrow{A_{route}} = |\theta_1 - \theta_2| \quad (14)$$

The agent's updated position in the next iteration is then given by:

$$A_i^{t+1} = A_i^t + rand * \overrightarrow{A_{route}} * (U_b - L_b) \quad (15)$$

Substituting **Eq. (14)** into **Eq. (15)**, we get the final update formula:

$$A_i^{t+1} = A_i^t + rand * |\theta_1 - \theta_2| * (U_b - L_b) \quad (16)$$

This formulation ensures that each agent dynamically adjusts its direction and range of movement based on the current landscape of solutions. The analogy of a direct-selling strategy emphasizes agility, precision, and contextual awareness, enabling agents to identify and pursue the most promising areas of the search space, just as early agents strategically identified the best routes and communities during their migration.

3.2.5. Action Control Mechanism

In the HSO algorithm, agents perform two types of movements: open movement and secret movement. The open movement represents the initial exploration phase, where agents migrate freely to discover promising areas in the search space. As the *hijrah* process progresses, agents switch to secret movement, which consists of two strategies: following a guide and taking an unusual route.

To control the transition between these strategies, the action control mechanism is applied. This mechanism uses an action control function $c_{f(t)}$ and a threshold constant $c_{0(t)}$. The function $c_{f(t)}$ generates random values fluctuating between 0 and 1 over time, as shown in **Eq. (17)** and **Fig. 5**. When $c_{f(t)}$ exceeds $c_{0(t)}$, agents perform open movement, focusing on

safe and directed search. When $c_{f(t)}$ falls below $c_{0(t)}$, they shift to secret movement, allowing for deeper exploitation of alternative paths. Since the exact $c_{0(t)}$ is unknown, it is set at 0.5, the midpoint between zero and one.

$$c_{f(t)} = \left| 2 * r_{[0,1]} - 1 * \left(1 - \frac{iter}{max_iter} \right) \right| \quad (17)$$

Here, $iter$ represents the i -th iteration of the algorithm cycle, and max_iter is the maximum number of iterations, which is an initialized parameter.

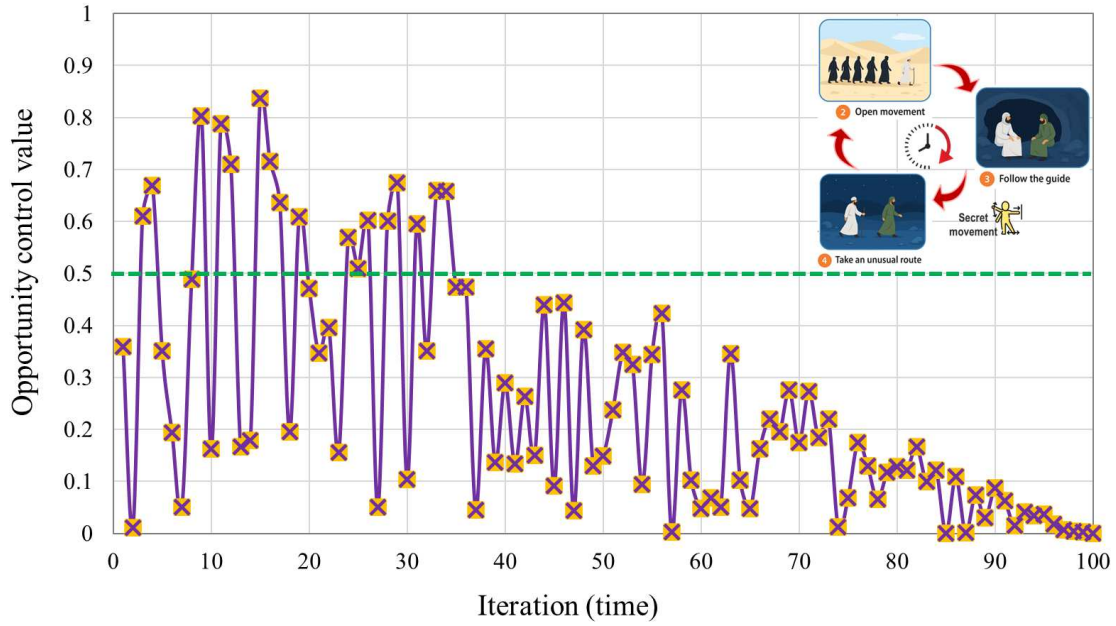


Fig. 5. Simulation of action control mechanism

3.2.6. Schematic Representation of HSO

The core principles and mathematical framework of HSO are presented in **Figs. 6** and **7**, respectively. During initialization, the agent population is generated using **Eq. (1)**. An action control mechanism, $c_{f(t)}$, is introduced to govern the transition between the open movement and secret movement phases, as defined in **Eq. (17)**. The agents' new positions are updated using **Eq. (5)** for open movement, **Eq. (9)** for following a guide during secret movement, and **Eq. (16)** for taking an unusual route in secret movement. The *hijrah* search in the field area operates in a cyclic manner until the current iteration count ($iter$) reaches the predefined maximum (max_iter). HSO employs NP parameter vectors in a D-dimensional search space.

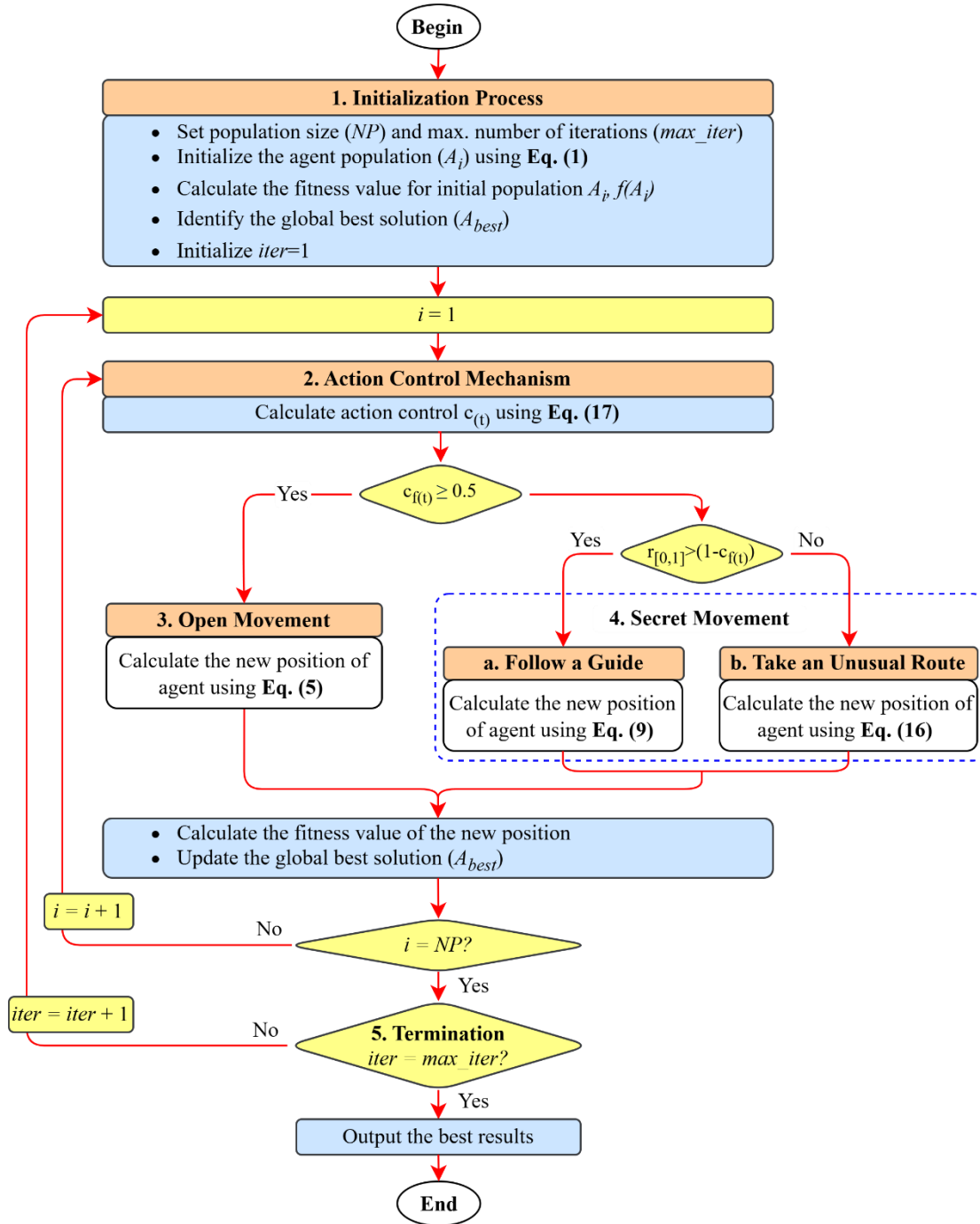


Fig. 6. Schematic flowchart of HSO algorithm

Algorithm: Hijrah Search Optimizer (HSO)

```
1: BEGIN
2:   % 1. Initialization Process
3:   Set population size (NP) and maximum number of iterations (max_iter)
4:   Initialize the agent population ( $A_i$ ) using Eq. (1)
5:   Calculate the fitness value for initial population  $A_i$ ,  $f(A_i)$ 
6:   Identify the global best solution ( $A_{best}$ )
7:   Initialize iter = 1
8:   REPEAT
9:     For  $i=1:NP$  do
10:      % 2. Action Control Mechanism
11:      Calculate action control ( $c_{f(t)}$ ) using Eq. (17)
12:      If  $c_{f(t)} \geq 0.5$ : 3. Open Movement
13:        Calculate the new position of agent using Eq. (5)
14:      Else: 4. Secret Movement
15:        If  $r_{[0,1]} > (1 - c_m)$ : a. Follow a Guide
16:          Calculate the new position of agent using Eq. (9)
17:        Else: b. Take an Unusual Route
18:          Calculate the new position of agent using Eq. (16)
19:        End if
20:      End if
21:      Calculate the fitness value of the new position
22:      Update the global best solution ( $A_{best}$ )
23:    End for i
24:    Update the iteration: iter = iter + 1
25:  UNTIL 5. Iteration: the stop criteria are met (e.g., iter > max_iter)
26:  Output the best results
27: END
```

Fig. 7. Pseudocode of the HSO algorithm

4. Computational Experiment

In this study, the performance of HSO was validated on several benchmark functions and compared with other metaheuristic algorithms.

4.1 Mathematical Benchmark Problems

4.1.1 Thirty Benchmark Functions

The performance of HSO was validated using a standard set of benchmark functions with known global optima, enabling objective evaluation and comparison with other algorithms. In this study, HSO was tested on a carefully selected set of thirty benchmark functions sourced from the literature [43], with detailed descriptions presented in **Table 2**.

Table 2. Thirty benchmark functions

No.	Function	Opt.	D	Range [Ub, Lb]	Type
1	SumSquares	0	30	[-10, 10]	Unimodal separable
2	Quartic	0	30	[-1.28, 1.28]	Unimodal separable
3	Beale	0	2	[-4.5, 4.5]	Unimodal non-separable
4	Easom	-1	2	[-100, 100]	Unimodal non-separable
5	Schwefel 1.2	0	30	[-100, 100]	Unimodal non-separable
6	Rosenbrock	0	30	[-30, 30]	Unimodal non-separable
7	Dixon–Price	0	30	[-10, 10]	Unimodal non-separable
8	Foxholes	0.998	2	[-65.536, 65.536]	Multimodal separable
9	Rastrigin	0	30	[-5.12, 5.12]	Multimodal separable
10	Michalewicz 2	-1.8013	2	[0, π]	Multimodal separable
11	Michalewicz 5	-4.6877	5	[0, π]	Multimodal separable
12	Michalewicz 10	-9.6602	10	[0, π]	Multimodal separable
13	Schaffer	0	2	[-100, 100]	Multimodal non-separable
14	Six-Hump Camel	-1.0316	2	[-5, 5]	Multimodal non-separable
15	Bohachevsky 2	0	2	[-100, 100]	Multimodal non-separable
16	Bohachevsky 3	0	2	[-100, 100]	Multimodal non-separable
17	Shubert	-186.7309	2	[-10, 10]	Multimodal non-separable
18	Goldstein–Price	3	2	[-2, 2]	Multimodal non-separable
19	Kowalik	0.000307	4	[-5, 5]	Multimodal non-separable
20	Shekel 5	-10.1532	4	[0, 10]	Multimodal non-separable
21	Shekel 7	-10.4029	4	[0, 10]	Multimodal non-separable
22	Shekel 10	-10.5364	4	[0, 10]	Multimodal non-separable
23	PowerSum	0	4	[0, 1]	Multimodal non-separable
24	Hartman 6	-3.3224	6	[0, 1]	Multimodal non-separable
25	Griewank	0	30	[-600, 600]	Multimodal non-separable
26	Ackley	0	30	[-32, 32]	Multimodal non-separable
27	Penalized	0	30	[-50, 50]	Multimodal non-separable
28	Langermann 10	NA	10	[0, 10]	Multimodal non-separable
29	Fletcher–Powell 2	0	2	$[-\pi, \pi]$	Multimodal non-separable
30	Fletcher–Powell 10	0	10	$[-\pi, \pi]$	Multimodal non-separable

Note: Opt. = optimum value, D = dimension, Ub = upper bound, Lb = lower bound

4.1.2 CEC-2020 Test Suite

To further evaluate the performance of HSO and its competing algorithms, the CEC-2020 test suite is utilized. This suite consists of ten diverse test functions, classified into four categories: unimodal, multimodal, hybrid, and composition, as outlined in **Table 3** [44].

Table 3. CEC-2020 test suite

No.	Function	Opt.	Type
CF1	Shifted and Rotated Bent Cigar	100	Unimodal function
CF2	Shifted and Rotated Schwefel's	1100	Multimodal shifted and rotated function
CF3	Shifted and Rotated Lunacek bi-Rastrigin	700	Multimodal shifted and rotated function
CF4	Expanded Rosenbrock's plus Griewangk's	1900	Multimodal shifted and rotated function
CF5	Hybrid function 1 (N = 3)	1700	Hybrid function (H)
CF6	Hybrid function 2 (N = 4)	1600	Hybrid function (H)
CF7	Hybrid function 3 (N = 5)	2100	Hybrid function (H)
CF8	Composition function 1 (N = 3)	2200	Composition function (C)
CF9	Composition function 1 (N = 4)	2400	Composition function (C)
CF10	Composition function 1 (N = 5)	2500	Composition function (C)

4.2 Performance Comparison

4.2.1 Evaluation Methods

Several evaluation methods were employed in this study to assess the performance of HSO in comparison with benchmark algorithms. One such method is the hit rate, defined in **Eq. (18)** as the ratio of successful attempts that reach the global optimum to the total number of independent optimization runs [45]. This metric reflects the algorithm's success rate in solving benchmark functions:

$$\text{Hit rate} = \frac{\sum \text{Achieve global optimum}}{\sum \text{Functions}} \quad (18)$$

All experiments were conducted using MATLAB R2024b on a PC equipped with an Intel(R) Core(TM) i5-8400 CPU (2.80 GHz), 16.0 GB RAM, and running Windows 11.

To further analyze performance differences between HSO and the comparison algorithms, non-parametric statistical tests were employed, specifically, the Wilcoxon rank-sum test and the Friedman mean rank test [46, 47]. These tests are commonly used to determine statistically significant differences among algorithms across multiple datasets.

The Wilcoxon rank-sum test, conducted at a 5% significance level ($\alpha = 0.05$), evaluates the null hypothesis (H_0) that there is no significant difference between the results of two algorithms ($p_{value} > \alpha$) against the alternative hypothesis (H_1) that HSO performs significantly better ($p_{value} \leq \alpha$):

$$\begin{aligned} H_0 : \mu_1 &\text{ is equivalent to } \mu_2 \\ H_1 : \mu_1 &\text{ is superior to } \mu_2 \end{aligned} \quad (19)$$

Additionally, the Friedman test statistic is computed using **Eq. (20)**:

$$X^2 = \frac{12}{N(k+1)} \left(\sum_{j=1}^k T_j^2 - \frac{k(k+1)^2}{4} \right) \quad (20)$$

where X^2 denotes the Friedman test statistic, N is the number of problems (subjects), k is the number of algorithms (treatments), and T_j is the sum of ranks for the j -th algorithm.

4.2.2 Comparative Performance of Algorithms

To ensure fair comparisons and prevent misleading conclusions, all algorithms were evaluated using an equal number of fitness function evaluations [48]. The stopping criterion was based on a commonly adopted approach, defined by the maximum number of evaluations, calculated as the product of the population size, number of phases, and number of iterations. Following the guideline established by Karaboga and Bahriye [49], the thirty benchmark functions used in this study were evaluated with a maximum of 500,000 function evaluations.

For the CEC-2020 benchmark suite, a population size of 50 was employed, with each function evaluated up to 10,000 times. To maintain consistency and fairness, all algorithms were tested under identical experimental conditions, with their respective internal parameter settings detailed in **Table 4**.

Table 4. Parameter settings used for the compared algorithms

No.	Year	Algorithm	Setting parameter
1	2026	Hijrah Search Optimizer (HSO)	NP, NE
2	2025	Birds-of-Paradise Search (BPS)	NP, NE Scaling factor in long distance movement (γ) = 0.1, scaling factor in gathering (ω) = 3
3	2020	Sparrow Search Algorithm (SSA)	NP, NE Producers accounts (P) = 0.2
4	2016	Whale Optimization Algorithm (WOA)	NP, NE Fluctuation range (a): decreased from 2 to 0, Coefficient of the logarithmic spiral shape (b) = 1
5	2014	Grey Wolf Optimizer (GWO)	NP, NE A was linearly decreased from 2 to 0
6	2011	Teaching-Learning-Based Optimization (TLBO)	NP, NE
7	2007	Artificial Bee Colony (ABC)	NP, NE Trial limit = Population size*dimensions
8	1995	Particle Swarm Optimization (PSO)	NP, NE Cognitive component = 1.8, Social component = 1.8, Inertia weight = 0.6

Note: population size (NP), number of evaluations (NE)

The performance of each algorithm on the thirty benchmark functions and the CEC-2020 test suite is summarized in **Tables 5** and **6**, respectively. The reported metrics, mean, best, and standard deviation, were obtained from 30 independent runs, in accordance with the evaluation protocol established in previous research [49].

Table 5. Comparison of HSO and other algorithms on thirty benchmark functions

Fun.	Index	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
F1	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.10E+00	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.06E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.89E+00	0.00E+00
F2	Mean	4.85E-06	3.19E-05	3.59E-04	1.21E-03	5.22E-03	8.54E-04	1.76E+00	9.74E-05
	Best	1.82E-06	3.75E-05	1.56E-04	7.73E-05	7.02E-04	4.04E-04	9.39E-01	1.07E-06
	Std.	1.72E-04	2.75E-05	1.38E-04	1.29E-03	4.93E-04	3.33E-04	5.10E-01	1.44E-04
F3	Mean	0.00E+00	0.00E+00	0.00E+00	2.54E-02	0.00E+00	0.00E+00	1.85E-04	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.36E-06	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	1.39E-01	0.00E+00	0.00E+00	1.67E-04	0.00E+00
F4	Mean	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-7.26E-01	-1.00E+00
	Best	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-9.96E-01	-1.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	2.53E-08	0.00E+00	0.00E+00	2.27E-01	0.00E+00
F5	Mean	0.00E+00	0.00E+00	0.00E+00	2.36E-08	3.66E-05	0.00E+00	1.21E+03	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	1.70E-09	0.00E+00	0.00E+00	2.87E+02	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	2.99E-08	2.04E-03	0.00E+00	6.98E+02	0.00E+00
F6	Mean	1.13E+00	2.18E+01	1.59E+01	2.70E+01	2.15E+01	1.87E+01	4.80E+03	6.19E+04
	Best	8.66E-02	2.07E+01	1.42E+01	2.65E+01	2.44E+01	1.69E+01	1.36E+03	3.58E+03
	Std.	1.18E+00	5.69E-01	1.08E+00	3.09E-01	3.02E-01	1.05E+00	3.72E+03	9.15E+04

Fun.	Index	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
F7	Mean	1.48E-01	6.67E-01	6.67E-01	6.67E-01	6.67E-01	6.67E-01	7.60E+01	0.00E+00
	Best	2.40E-02	6.67E-01	6.67E+01	6.67E-01	6.67E-01	6.67E-01	2.17E+01	0.00E+00
	Std.	1.46E-01	9.06E-13	0.00E+00	7.05E-05	3.22E-05	2.47E-16	6.71E+01	0.00E+00
F8	Mean	9.98E-01	9.98E-01	9.98E-01	1.23E+00	9.98E-01	9.98E-01	9.98E-01	1.76E+00
	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Std.	0.00E+00	1.3E-16	5.83E-17	5.64E-01	4.23E-13	9.22E-17	2.33E-11	1.46E+00
F9	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.51E+00	7.39E+01	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.57E+01	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.14E+00	7.77E+00	0.00E+00
F10	Mean	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E-00	-1.80E-00	-1.80E+00	-1.80E+00	-1.80E+00
	Best	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00
	Std.	0.00E+00	0.00E+00	9.03E-16	7.50E-08	4.52E-07	9.03E-16	1.02E-15	9.03E-16
F11	Mean	-4.69E+00	-4.69E+00	-4.69E+00	-3.81E+00	-3.01E+00	-4.63E+00	-4.69E+00	-4.59E+00
	Best	-4.69E+00	-4.69E+00	-4.69E+00	-4.65E+00	-4.65E+01	-4.69E+00	-4.69E+00	-4.69E+00
	Std.	2.95E-05	4.41E-10	1.95E-15	6.04E-01	3.04E-02	7.41E-02	8.90E+00	1.57E-01
F12	Mean	-9.66E+00	-9.55E+00	-9.58E+00	-6.39E+00	-6.45E+00	-9.35E+00	-9.06E+00	-8.73E+00
	Best	-9.66E+00	-9.64E+00	-9.66E+00	-8.66E+00	-8.69E+00	-9.63E+00	-9.31E+00	-9.43E+00
	Std.	1.45E-01	3.31E-02	1.11E-01	1.11E+00	3.12E-01	2.16E-01	1.25E-01	5.63E-01
F13	Mean	0.00E+00	0.00E+00	0.00E+00	1.75E-02	0.00E+00	0.00E+00	8.93E-05	1.46E-03
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.16E-07	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	2.17E-02	0.00E+00	0.00E+00	1.67E-04	7.97E-03
F14	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std.	0.00E+00	0.00E+00	6.78E-16	2.89E-11	3.03E-13	6.78E-16	2.29E-10	6.58E-16
F15	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.87E-10	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.72E-12	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.10E-10	0.00E+00
F16	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.93E-05	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.35E-06	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	2.58E-16	0.00E+00	0.00E+00	3.86E-05	0.00E+00
F17	Mean	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02
	Best	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02
	Std.	0.00E+00	0.00E+00	2.45E-05	3.03E-05	2.83E-07	2.42E-16	8.23E-04	3.70E-14
F18	Mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Best	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Std.	5.53E-16	1.22E-15	0.00E+00	3.07E-06	8.32E-10	1.22E-15	1.29E-03	5.34E-16
F19	Mean	3.10E-04	3.38E-04	3.07E-04	6.92E-04	6.05E-04	3.73E-03	1.16E-03	5.99E-04
	Best	3.10E-04	3.07E-04	3.07E-04	3.09E-04	3.11E-04	3.07E-03	8.00E-04	3.07E-04
	Std.	1.30E-04	2.97E-12	4.49E-14	3.70E-04	3.29E-04	1.68E-04	1.98E-04	3.61E-04
F20	Mean	-1.02E+01	-1.02E+01	-9.81E+00	-8.80E+00	-8.32E-00	-1.00E+01	-1.01E+01	-5.07E+00
	Best	-1.02E+01	-1.02E+01	1.29E+00	-1.02E+01	-1.23E+01	-1.00E+01	-1.02E+02	-1.01E+01
	Std.	6.85E-15	4.82E-13	-1.01E+01	2.55E+00	1.04E+00	6.74E-15	1.50E-02	3.24E+00
F21	Mean	-1.04E+01	-1.04E+01	-1.02E+01	-9.43E+00	-9.04E+00	-1.04E+01	-1.04E+01	-6.20E+00
	Best	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-3.00E+01	-1.04E+01	-1.04E+01	-1.04E+01
	Std.	2.27E-15	3.56E-10	9.70E-01	2.24E+00	1.44E+00	1.04E-15	1.50E-02	3.58E+00
F22	Mean	-1.05E+01	-1.05E+01	-1.05E+01	-9.69E+00	-9.00E+00	-1.05E+01	-1.05E+01	-5.15E+00
	Best	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-2.01E+01	-1.05E+01	-1.05E+01	-1.06E+01
	Std.	1.81E-15	6.01E-10	1.75E-15	2.21E+00	1.09E+00	1.89E-11	1.69E-02	3.41E+00
F23	Mean	1.99E-03	8.75E-04	3.81E-04	1.45E+00	1.92E-01	1.15E-02	1.27E-02	1.23E-03
	Best	0.00E+00	6.92E-07	6.87E-06	3.75E-02	2.72E-02	7.96E-03	1.47E-03	5.10E-09
	Std.	2.29E-03	1.66E-03	5.69E-04	1.75E+00	9.02E-01	1.51E-03	9.61E-03	3.01E-03
F24	Mean	-3.32E+00	-3.32E+00	-3.29E+00	-3.23E+00	-3.23E+00	-3.32E+00	-3.32E+00	-3.25E+00
	Best	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
	Std.	8.88E-16	1.26E-10	5.56E-02	9.45E-02	8.02E-02	1.32E-02	4.11E-05	6.01E-02
F25	Mean	0.00E+00	0.00E+00	0.00E+00	4.51E-03	3.11E-03	7.00E-10	1.56E+00	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.03E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	2.47E-02	2.41E-02	0.00E+00	2.95E-01	0.00E+00
F26	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.52E-02	9.97E+00	7.65E+00
	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.56E+00	4.84E+00

Fun.	Index	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
	Std.	2.02E-18	0.00E+00	0.00E+00	2.76E-15	3.89E-15	3.01E-01	1.13E+00	1.31E+00
F27	Mean	3.54E-03	6.80E-08	0.00E+00	2.91E-03	3.12E-04	0.00E+00	1.27E+00	1.55E+01
	Best	2.29E-06	1.35E-09	0.00E+00	3.17E-04	2.02E-04	0.00E+00	3.44E-01	3.06E+00
	Std.	1.89E-02	1.07E-07	0.00E+00	4.22E-03	3.92E-03	6.13E-17	4.97E-01	1.33E+01
F28	Mean	-7.98E-01	-6.80E-01	-7.27E-01	-3.12E-01	-2.09E-01	-7.45E-01	-4.87E-01	-4.81E-01
	Best	-1.50E+00	-1.50E+00	-1.50E+00	-7.98E-01	-8.08E-01	-1.50E+00	-7.98E-01	-7.98E-01
	Std.	1.08E-03	2.59E-02	3.40E-01	2.03E-01	3.33E-01	4.17E-01	1.54E-01	1.96E-01
F29	Mean	0.00E+00	0.00E+00	0.00E+00	1.95E-06	1.05E-05	0.00E+00	2.25E-01	0.00E+00
	Best	0.00E+00	0.00E+00	0.00E+00	3.45E-10	2.55E-10	0.00E+00	3.82E-05	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	4.76E-06	3.09E-06	1.14E-22	4.12E-01	0.00E+00
F30	Mean	0.00E+00	1.31E-01	2.09E-02	7.67E+02	1.67E+01	2.07E+01	3.44E+00	7.60E+02
	Best	0.00E+00	0.00E+00	0.00E+00	5.67E-02	1.22E+00	0.00E+00	2.50E-02	1.21E-07
	Std.	5.00E-05	2.67E-01	1.14E-01	1.35E+03	1.05E+01	2.25E+01	6.66E+00	1.55E+03

Note: Std. = standard deviation

Table 6. Comparison of HSO and other algorithms on the CEC-2020 test suite

Fun.	Index	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
CF1	Mean	2.87E+08	3.28E+09	7.52E+08	7.09E+08	3.98E+09	1.26E+10	2.27E+09	2.38E+10
	Best	2.77E+08	3.21E+09	7.42E+08	7.08E+08	3.32E+09	9.29E+09	2.19E+09	2.20E+10
	Std.	9.45E+07	2.32E+08	2.28E+07	1.53E+08	3.32E+08	3.82E+09	1.28E+07	3.27E+07
CF2	Mean	6.06E+03	5.97E+03	6.75E+04	8.28E+04	9.66E+03	3.29E+03	7.38E+03	7.37E+03
	Best	5.25E+03	4.02E+03	6.29E+04	8.01E+04	8.22E+03	2.92E+03	7.28E+03	7.14E+03
	Std.	5.43E+03	7.33E+02	7.03E+02	9.06E+03	1.18E+03	8.34E+02	5.38E+02	9.92E+02
CF3	Mean	9.53E+02	1.11E+03	6.99E+03	9.82E+02	6.62E+03	4.11E+03	1.24E+03	8.38E+02
	Best	9.02E+02	9.53E+02	6.39E+03	9.33E+02	6.56E+03	3.29E+03	1.22E+03	8.18E+02
	Std.	6.02E+02	6.53E+01	6.02E+02	7.03E+02	9.24E+02	1.34E+02	6.38E+02	6.22E+01
CF4	Mean	7.95E+03	7.26E+03	8.76E+03	3.27E+04	9.74E+03	1.29E+05	7.53E+04	1.38E+06
	Best	7.58E+03	7.12E+03	8.64E+03	3.14E+04	8.21E+03	1.21E+05	7.12E+04	1.23E+06
	Std.	5.92E+02	1.47E+04	3.22E+02	1.60E+04	2.88E+03	2.38E+03	6.82E+03	8.38E+04
CF5	Mean	6.94E+03	6.92E+05	7.29E+05	4.19E+04	6.13E+03	2.27E+06	7.27E+07	4.54E+05
	Best	5.03E+03	6.21E+05	7.14E+05	3.21E+04	5.27E+03	2.21E+06	7.07E+07	4.27E+05
	Std.	1.22E+03	6.02E+05	1.39E+05	8.75E+03	1.92E+03	1.72E+05	8.55E+05	9.39E+04
CF6	Mean	2.43E+03	1.95E+03	2.40E+03	5.64E+03	2.67E+03	6.28E+03	9.45E+03	4.29E+03
	Best	2.22E+03	1.93E+03	2.24E+03	5.57E+03	2.21E+03	5.42E+03	9.12E+03	4.02E+03
	Std.	9.00E+02	1.42E+02	1.28E+02	1.75E+03	6.93E+02	1.18E+02	7.32E+03	2.92E+02
CF7	Mean	1.02E+04	6.92E+04	6.54E+04	1.44E+04	9.30E+03	1.38E+05	9.48E+03	2.54E+05
	Best	9.92E+03	6.91E+04	6.43E+04	9.91E+03	9.09E+03	9.38E+04	9.28E+03	2.27E+05
	Std.	5.00E+03	2.33E+04	7.33E+03	7.11E+03	9.92E+02	2.27E+04	6.38E+03	1.48E+05
CF8	Mean	2.32E+03	2.30E+03	5.60E+04	4.93E+03	2.01E+04	9.31E+03	5.58E+03	4.69E+03
	Best	2.29E+03	2.19E+03	5.21E+04	4.28E+03	7.33E+03	9.18E+03	5.13E+03	4.06E+03
	Std.	1.82E+03	3.28E+02	5.90E+03	1.07E+03	8.78E+03	7.52E+02	1.28E+03	6.17E+02
CF9	Mean	8.01E+02	2.10E+03	8.29E+03	7.40E+03	9.66E+03	6.83E+03	8.28E+03	3.29E+03
	Best	8.00E+02	2.03E+03	8.19E+03	7.23E+03	9.19E+03	6.81E+03	8.27E+03	2.12E+03
	Std.	1.43E+02	2.56E+02	6.31E+02	1.90E+03	1.04E+03	1.26E+02	5.22E+03	7.34E+02
CF10	Mean	3.32E+03	2.08E+03	7.06E+03	2.29E+03	1.68E+04	4.28E+03	5.33E+04	4.91E+03
	Best	3.21E+03	2.80E+03	7.43E+03	2.12E+03	1.49E+04	4.01E+03	5.17E+04	4.10E+03
	Std.	7.22E+02	5.21E+02	6.54E+02	1.42E+03	2.19E+03	9.62E+02	3.34E+04	3.18E+03

Note: Std. = standard deviation

The mathematical test results presented in **Table 7** demonstrate the superior performance of the HSO algorithm, which achieved the best mean value in 26 out of 30 benchmark functions (86.67%). This performance surpassed all competing algorithms, including BPS and SSA, which reached optimal results in 24 (80.00%) and 22 (73.33%) functions, respectively. In cases where the optimal value was not attained, HSO still achieved the best or tied performance in

90.00% of the functions, compared to 86.67% for BPS and 73.33% for SSA. The Wilcoxon signed-rank test, used for pairwise comparisons, indicates that HSO outperformed (+) BPS in 4 functions, underperformed (-) in 1 function, and showed no statistically significant difference ($\approx$) in the remaining 25 functions. Furthermore, the Friedman test, which evaluates algorithm rankings across multiple problems, placed HSO in first position with a score of 2.63, ahead of BPS (2.74) and SSA (2.89). On the CEC-2020 test suite, HSO continued to demonstrate its advantage, outperforming its rivals in both the Wilcoxon and Friedman tests, as detailed in **Table 8**.

Table 7. Cumulative results: HSO vs. other algorithms on thirty benchmark functions

Perform.	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
Mean output	26	24	22	15	16	17	9	14
	86.67%	80.00%	73.33%	50.00%	53.33%	56.67%	30.00%	46.67%
Best output	27	26	22	19	17	20	10	16
	90.00%	86.67%	73.33%	63.33%	56.67%	66.67%	33.33%	53.33%
Wilcoxon test (+/-/ $\approx$)		4/1/25	7/1/22	13/0/17	14/0/16	9/1/20	20/0/10	13/1/16
Friedman test	2.63	2.74	2.89	7.35	5.72	4.08	8.95	5.70
Rank	1	2	3	7	6	4	8	5

Table 8. Cumulative results: HSO vs. other algorithms on CEC-2020

Perform.	HSO	BPS	SSA	WOA	GWO	TLBO	ABC	PSO
Wilcoxon test (+/-/ $\approx$)		5/5/0	9/1/0	9/1/0	8/2/0	9/1/0	9/1/0	9/1/0
Friedman test	3.98	5.20	7.54	6.78	7.89	8.48	9.18	3.98
Rank	1	2	4	3	6	7	8	1

This study also evaluated the convergence performance of the algorithms, emphasizing the need to balance exploration and exploitation to prevent premature convergence to local optima [50]. The convergence curves of HSO on unimodal and multimodal functions are presented in **Figs. 8** and **9**, where HSO consistently outperformed the other algorithms. On unimodal functions (F1, F7 in **Fig. 8**, and CF1 in **Fig. 9**), HSO efficiently exploited promising regions. For multimodal functions (F12, F22 in **Fig. 8**, and CF3 in **Fig. 9**), it successfully avoided being trapped in local optima. Moreover, for the hybrid function (CF7) and composition function (CF9), HSO achieved faster convergence than its competitors, as shown in **Fig. 9**.

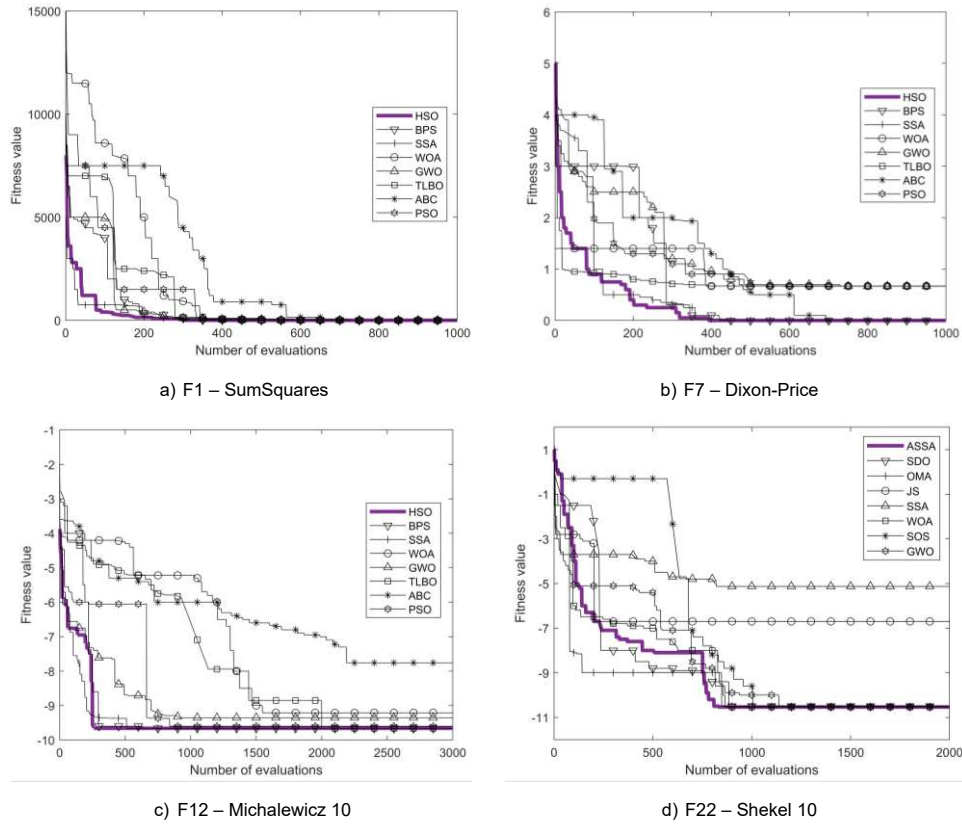


Fig. 8. Convergence curves for selected thirty benchmark functions: F1, F7, F12, and F22

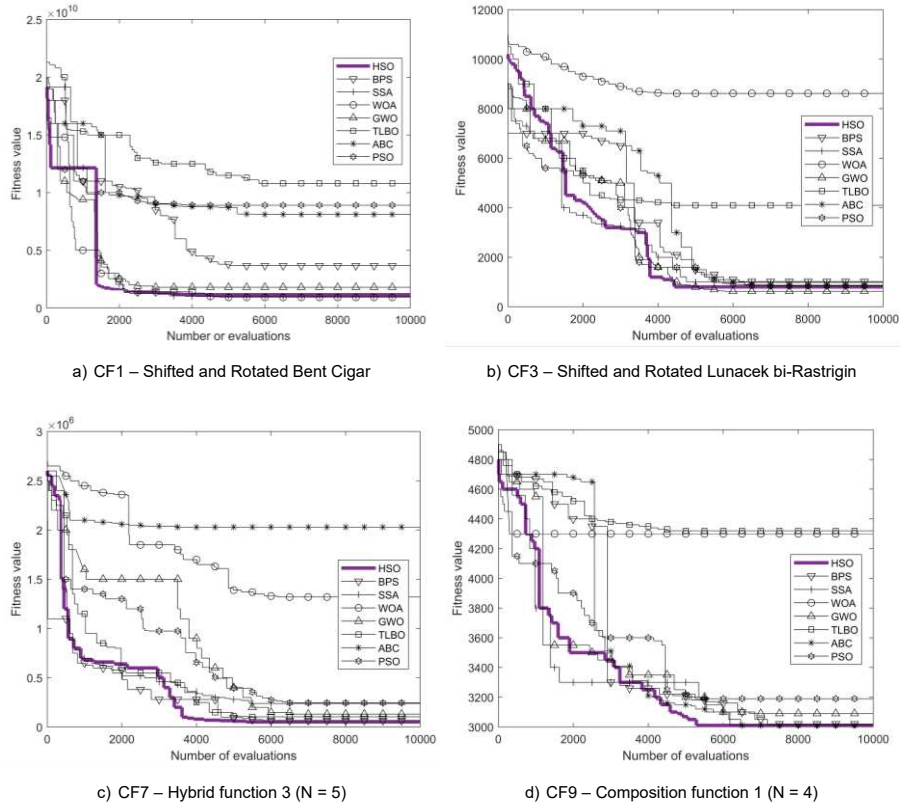


Fig. 9. Convergence curves for selected CEC-2020 tests: CF1, CF3, CF7, and CF9

Additionally, **Figs. 10** and **11** display stacked bar charts illustrating the ranking distribution of each algorithm across the thirty benchmark functions and the CEC-2020 test suite, respectively. These visualizations highlight HSO's strong overall performance, as it frequently secured top positions. Specifically, for the thirty benchmark functions, HSO ranked 1st in 26 functions, 2nd in 2, 4th in 1, and 6th in 1. In the CEC-2020 test suite, HSO achieved 1st place in 2 functions, 2nd in 4, and 3rd in 4.

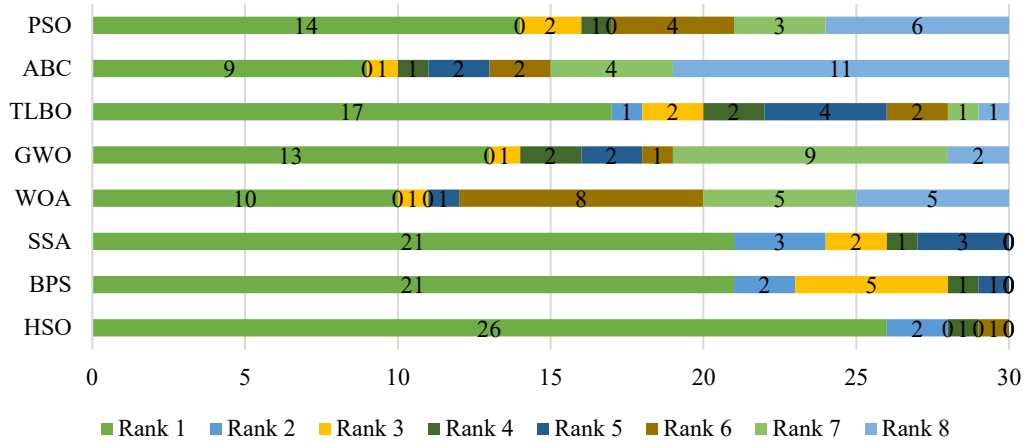


Fig. 10. Ranking distribution of competing algorithms across the thirty benchmark functions.

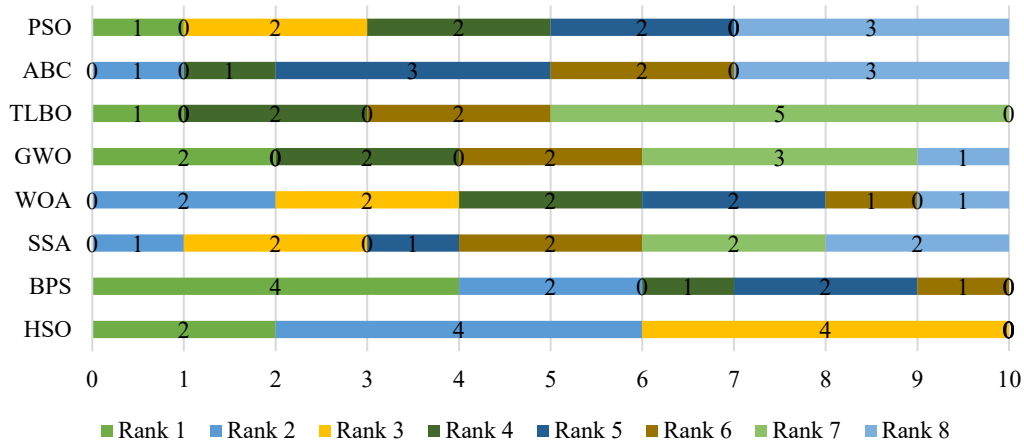


Fig. 11. Ranking distribution of competing algorithms on the CEC-2020 test suite

5. Engineering Design Problems

This section presents the application of the HSO algorithm to engineering design problems, specifically the welded beam design and the 52-bar tower structure.

5.1 Welded Beam Design (WBD)

The first case study involves the optimization of a welded beam design. The objective is to optimize the beam's dimensions—specifically, the thickness (h), length (l), height (t),

thickness (b), along with the weld size, to minimize the overall cost. **Fig. 12** illustrates the beam structure. Let $\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b]$. The mathematical formulations for the welded beam design problem are presented in **Eqs. (21), (22), and (23)**.

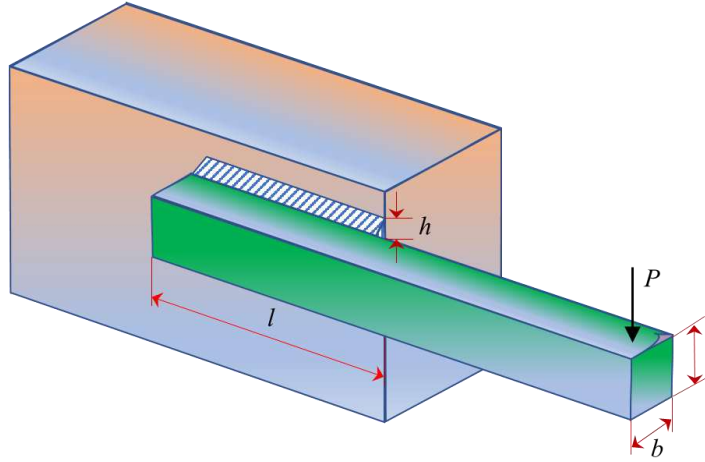


Fig. 12. Welded beam design

Minimize:

$$f(\vec{x}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \quad (21)$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0; g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0; g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0;$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0; g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0; g_6(\vec{x}) = 0.125 - x_1 \leq 0;$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0 \quad (22)$$

The parameter ranges are defined as follows: $0.1 \leq x_1$, $0.1 \leq x_2$, $x_3 \leq 10$, and $x_4 \leq 2$, with further details provided below.

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}; \tau' = \frac{p}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J};$$

$$M = P\left(L + \frac{x_2}{2}\right); R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}; J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\},$$

$$\sigma(\vec{x}) = \frac{6PL}{x_3^2x_4}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}; P_c(\vec{x}) = \frac{\sqrt{\frac{40x}{36}}}{L^2}\left(1 - \frac{x_3^2}{2L}\sqrt{\frac{E}{4G}}\right) \quad (23)$$

where the value of $P = 6,000$ lb, $\delta_{max} = 0.25$ in., $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{max} = 13,600$ psi, and $\sigma_{max} = 30,000$ psi.

As shown in **Table 9**, the HSO achieved the best solution for the welded beam design problem at 1.6701, followed by the Red-billed Blue Magpie Optimizer (RBMO) and Dung Beetle Optimizer (DBO). Statistical results also highlight the robustness of BPS, with a low standard deviation of 1.002×10^{-9} . Additionally, the zoomed-in convergence graph in **Fig. 13**

shows that HSO converges significantly faster to a near-optimal solution in the early iterations compared to BPS.

Table 9. Results comparison for the welded beam design problem

Algorithms	Variables				Mean	Worst	Best	Std.	Rank
	h	l	t	b					
ES [51]	0.2030	3.5310	9.0361	0.2057	2.6918	5.0625	1.7287	0.7409	12
EO [51]	0.2057	3.4705	9.0366	0.2057	1.7299	1.7874	1.7248	0.0119	6
QSA [51]	0.2057	3.4705	9.0366	0.2057	1.7248	1.7248	1.7248	0.0001	5
EBO [51]	0.2056	3.4739	9.0369	0.2057	2.0347	2.9153	1.7251	0.2124	11
ASO [51]	0.2036	3.5170	9.0382	0.2057	1.9402	2.3198	1.7279	0.1405	9
MFO [51]	0.2057	3.4705	9.0366	0.2057	1.9811	2.9547	1.7248	0.3259	10
MVO [51]	0.2057	3.4723	9.0428	0.2058	1.7504	1.8523	1.7266	0.0235	7
SSA [51]	0.2055	3.4666	9.0608	0.2056	1.8216	2.1836	1.7271	0.1006	8
DBO [52]	-	-	-	-	1.7149	1.8167	1.6702	0.0061	4
RBMO [52]	-	-	-	-	1.6702	1.6702	1.6835	9.72E-09	3
BPS	0.2057	3.4705	9.0366	0.2057	1.6701	1.6705	1.6701	1.00E-09	2
HSO	0.2057	3.4705	9.0366	0.2057	1.6700	1.6704	1.6700	7.52E-08	1

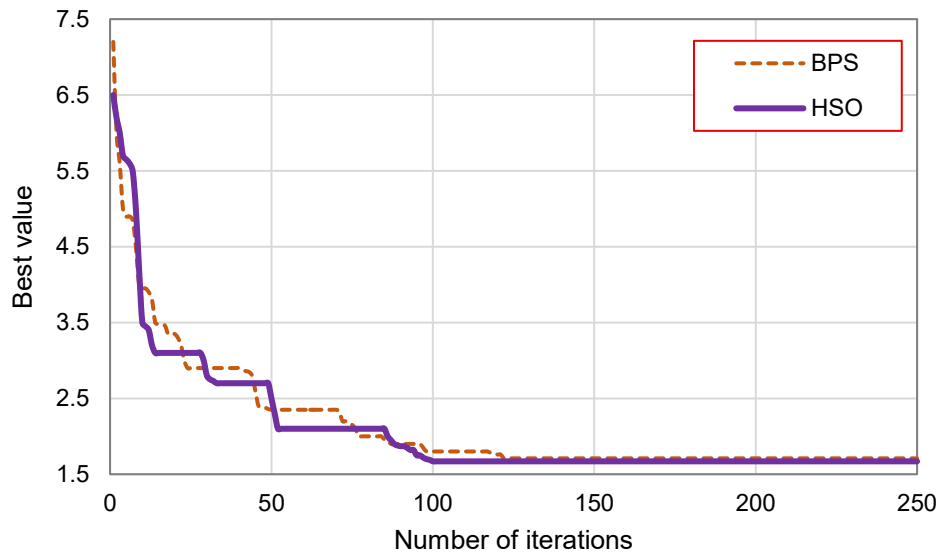


Fig. 13. HSO convergence trajectory for the welded beam design problem

5.2 52-Bar Tower

The second case study addresses the structural optimization of the 52-bar tower shown in **Fig. 14**, a problem previously tackled using algorithms like CSS [53], TLBO [54], AFA [55], and DAJA [55]. The structure's members are grouped into 12 categories: (a) A1-A4, (b) A5-A10, (c) A11-A13, (d) A14-A17, (e) A18-A23, (f) A24-A26, (g) A27-A30, (h) A31-A36, (i) A37-A39, (j) A40-A43, (k) A44-A49, and (l) A50-A52. Material properties include a density of 7860 kg/m³ and an elasticity modulus of 2.07 x 10⁵ MPa. Each member must satisfy a stress limit of ± 180 MPa, with loads $P_x = 100$ kN and $P_y = 200$ kN applied at nodes 17-20. Discrete variables follow AISC specifications, as used by Sadollah et al. [56] and Jawad et al. [57].

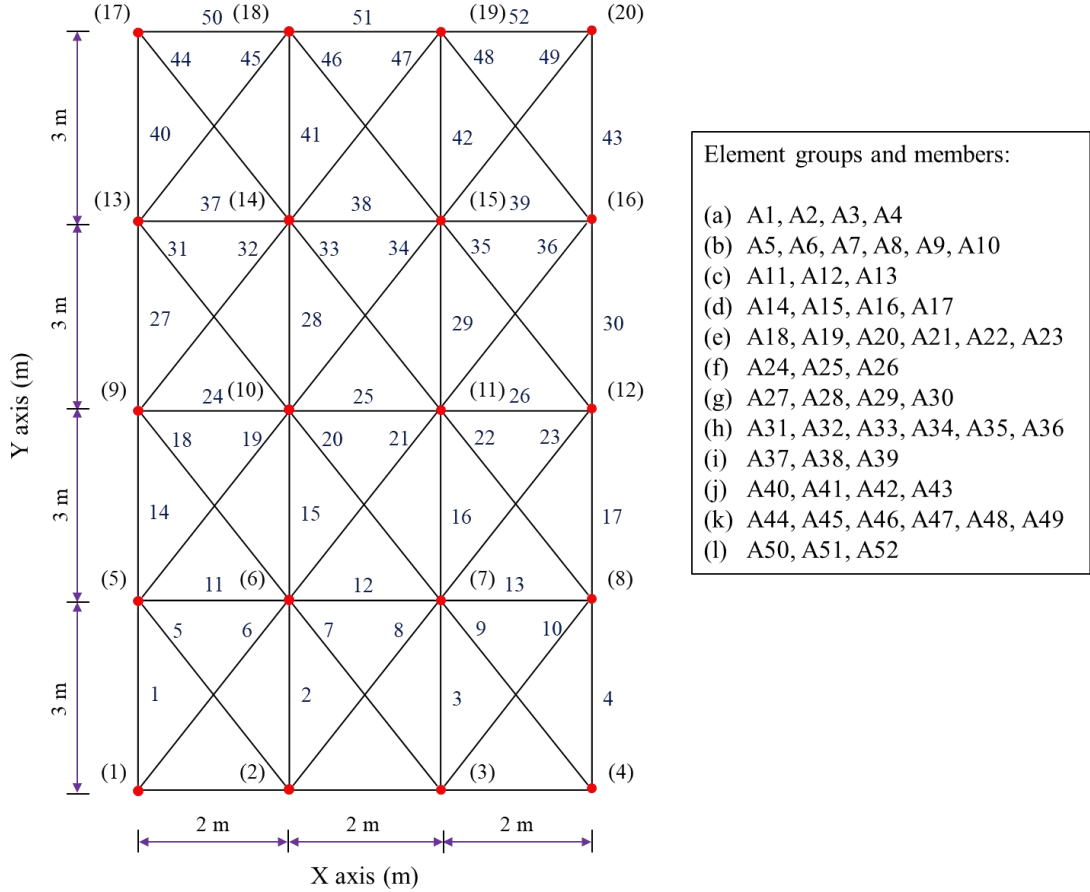


Fig. 14. 52-bar tower design

The objective of this problem is to minimize the structural weight or volume while ensuring that the constraints on element stresses and nodal displacements are satisfied [58]. For a structure consisting of nm elements (grouped into ng groups) and nn nodes, the problem can be formulated as follows:

Minimize:

$$W(A) = \sum_{i=1}^{nm} \gamma_i A_i \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2 + (z_{i1} - z_{i2})^2} \quad (24)$$

Subject to:

$$\sigma_i^c \leq \sigma_i \leq \sigma_i^t, \quad i = 1, 2, \dots, nm \quad (25)$$

$$\delta_{min} \leq \delta_j \leq \delta_{max}, \quad j = 1, 2, \dots, nm \quad (26)$$

$$A_{min} \leq A_k \leq A_{max}, \quad k = 1, 2, \dots, ng \quad (27)$$

where $W(A)$ denotes the weight of the structure; A is the vector of sizing variables, representing the cross-sectional areas of the bars, which are associated with the coordinates $x_{i1,2}$, $y_{i1,2}$, $z_{i1,2}$ of the nodes defining the i -th element; γ_i is the material density of member i ; and A_i is its cross-sectional area i , bounded between A_{min} and A_{max} . A penalty function F_p is introduced to enforce the stress and displacement constraints:

$$F_p = W(A) * (1 + \emptyset)^\varepsilon \quad (28)$$

where $\emptyset$ represents the combined penalty terms for stress and displacement constraints, defined as follows:

$$\emptyset = \sum_{i=1}^{nm} \emptyset_\sigma^i + \sum_{i=1}^{nm} \emptyset_\delta^j \quad (29)$$

The stress constraint penalty, $\emptyset_\sigma^i$, for the i -th member and the displacement constraint penalty, $\emptyset_\delta^j$, for the j -th node are defined as follows:

$$\begin{cases} \emptyset_\sigma^i = 0 & \text{if } \sigma_i^c \leq \sigma_i \leq \sigma_i^t \\ \emptyset_\sigma^i = \frac{|\sigma_i - \sigma_i^{t,c}|}{|\sigma_i^{t,c}|} & \text{if } \sigma_i(\sigma_i^c \text{ or } \sigma_i)\sigma_i^t \end{cases} \quad (30)$$

$$\begin{cases} \emptyset_\delta^j = 0 & \text{if } \delta_{min} \leq \delta_j \leq \delta_{max} \\ \emptyset_\delta^j = \frac{|\delta_j - \delta_{min,max}|}{|\delta_{min,max}|} & \text{if } \delta_j(\delta_{min} \text{ or } \delta_j)\delta_{max} \end{cases} \quad (31)$$

The exponent ε in the penalty function F_p varies with the iteration count and is defined as follows:

$$\varepsilon = \varepsilon_0 + \left(1 + \frac{iter}{\max_iter}\right) \quad (32)$$

where ε_0 denotes the initial value, which can be selected within the range of 1.001 to 10,000 [55]. In this study, $\varepsilon_0 = 2$ is adopted. Here, *iter* represents the current iteration number, and *max_iter* refers to the maximum number of iterations.

The proposed HSO algorithm is governed by two primary parameters: population size and the number of iterations. In this study, the population size was set to 40. The number of iterations serves as the main stopping criterion and can be assigned a high value (e.g., 1000 iterations). An additional termination condition is applied by monitoring the objective function, stopping the process if ten consecutive changes fall below 10^{-12} . As presented in **Table 10**, the three best optimal structural masses were obtained by HSO (1897.618 kg), CSS (1897.620 kg), and BPS (1899.685 kg). Among these, HSO required the fewest structural analyses (3,725), compared to CSS (5,000) and BPS (3,950). Thus, HSO demonstrates superior efficiency in solving this problem.

451

Table 10. Comparison of HSO and competitors for 52-bar tower problem

Design variables (mm ²)	CSS [53]	TLBO [54]	AFA [55]	DAJA [55]	BPS	HSO
$A_1 - A_4$	4658.055	4568.055	4658.055	4658.055	4568.055	4568.055
$A_5 - A_{10}$	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
$A_{11} - A_{13}$	388.386	494.193	363.225	494.193	494.193	494.193
$A_{14} - A_{17}$	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
$A_{18} - A_{23}$	940.000	940.000	939.988	940.000	940.000	939.998
$A_{24} - A_{26}$	494.193	494.193	494.193	494.193	494.193	506.451
$A_{27} - A_{30}$	2238.705	2238.705	2238.705	2238.705	2238.705	2238.705
$A_{31} - A_{36}$	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
$A_{37} - A_{39}$	494.193	494.193	641.289	494.193	494.193	388.386
$A_{40} - A_{43}$	1283.868	1283.686	1283.868	1283.868	1283.868	1283.868
$A_{44} - A_{49}$	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
$A_{50} - A_{52}$	494.193	494.193	494.193	494.193	494.193	506.451
Mean weight (kg)	N/A	N/A	N/A	1902.605	1906.735	1902.142
Worst weight (kg)	N/A	N/A	N/A	1903.944	1925.714	1904.991
Best weight (kg)	1897.620	1902.605	1903.370	1902.605	1902.605	1897.618
Std. (kg)	N/A	N/A	N/A	0.446	6.679	1.860
No. structure analyses	5,000	6,000	52,600	3,321	3,950	3,725

452 **6. Conclusion**

453 While the proposed HSO algorithm exhibits notable strengths and delivers impressive
454 results, it is not without limitations. First, due to time and resource constraints, its applicability
455 to real-world scenarios was assessed using only two case studies. Broader validation across
456 diverse domains is necessary to fully establish the effectiveness of HSO. Second, although this
457 study employed thirty benchmark functions and the CEC-2020 test suite to compare HSO with
458 eight existing models, future research could extend this evaluation by incorporating additional
459 benchmark functions and more advanced algorithms for a more comprehensive performance
460 assessment.

461 **Declarations**

462 **Conflict of interest:** The authors declare that there is no conflict of interest.

463 **References**

- 464 [1] A. Sinha, D. Pujara, and H. K. Singh, "Decomposition of Difficulties in Complex
465 Optimization Problems using a Bilevel Approach," in *2024 IEEE Congress on
466 Evolutionary Computation (CEC)*, 2024: IEEE, pp. 1-8, doi:
467 <https://doi.org/10.1109/CEC60901.2024.10611939>.
- 468 [2] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer
469 for solving engineering problems," *Expert Systems with Applications*, vol. 166, p.
470 113917, 2021, doi: <https://doi.org/10.1016/j.eswa.2020.113917>.
- 471 [3] J. Qiao *et al.*, "A hybrid particle swarm optimization algorithm for solving engineering
472 problem," *Scientific Reports*, vol. 14, no. 1, p. 8357, 2024, doi:
473 <https://doi.org/10.1038/s41598-024-59034-2>.

- [4] M. A. Şahman, "A discrete spotted hyena optimizer for solving distributed job shop scheduling problems," *Applied Soft Computing*, vol. 106, p. 107349, 2021, doi: <https://doi.org/10.1016/j.asoc.2021.107349>.
- [5] B. Toaza and D. Esztergár-Kiss, "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems," *Applied Soft Computing*, vol. 148, p. 110908, 2023, doi: <https://doi.org/10.1016/j.asoc.2023.110908>.
- [6] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16591-16633, 2023, doi: <https://doi.org/10.1007/s11042-022-13820-0>.
- [7] M. Abdel-Salam, A. I. Alzahrani, F. Alblehai, R. A. Zitar, and L. Abualigah, "An improved Genghis Khan optimizer based on enhanced solution quality strategy for global optimization and feature selection problems," *Knowledge-Based Systems*, vol. 302, p. 112347, 2024, doi: <https://doi.org/10.1016/j.knosys.2024.112347>.
- [8] Y. Wang, S. Ran, and G.-G. Wang, "Role-oriented binary grey wolf optimizer using foraging-following and Lévy flight for feature selection," *Applied Mathematical Modelling*, vol. 126, pp. 310-326, 2024, doi: <https://doi.org/10.1016/j.apm.2023.08.043>.
- [9] A. Qi *et al.*, "Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation," *Computers in biology and medicine*, vol. 148, p. 105810, 2022, doi: <https://doi.org/10.1016/j.combiomed.2022.105810>.
- [10] L. Wang, Z. Pan, and J. Wang, "A review of reinforcement learning based intelligent optimization for manufacturing scheduling," *Complex System Modeling and Simulation*, vol. 1, no. 4, pp. 257-270, 2021, doi: <https://doi.org/10.23919/CSMS.2021.0027>.
- [11] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13187-13257, 2023, doi: <https://doi.org/10.1007/s10462-023-10470-y>.
- [12] T. Dokeroglu, E. Sevinc, T. Kucukylmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 137, p. 106040, 2019, doi: <https://doi.org/10.1016/j.cie.2019.106040>.
- [13] J. J. Jui, M. I. Molla, M. A. Ahmad, and I. T. Hettiarachchi, "Recent Advances and Applications of the Multi-verse Optimiser Algorithm: A Survey from 2020 to 2024," *Archives of Computational Methods in Engineering*, pp. 1-34, 2025, doi: <https://doi.org/10.1007/s11831-025-10277-w>.
- [14] M. A. El-Shorbagy, A. Bouaouda, L. Abualigah, and F. A. Hashim, "Atom Search Optimization: a comprehensive review of its variants, applications, and future directions," *PeerJ Computer Science*, vol. 11, p. e2722, 2025, doi: <https://doi.org/10.7717/peerj-cs.2722>.
- [15] R. Rai, K. G. Dhal, A. Das, and S. Ray, "An inclusive survey on marine predators algorithm: Variants and applications," *Archives of Computational Methods in Engineering*, vol. 30, no. 5, pp. 3133-3172, 2023, doi: <https://doi.org/10.1007/s11831-023-09897-x>.
- [16] L. Velasco, H. Guerrero, and A. Hospitaler, "A literature review and critical analysis of metaheuristics recently developed," *Archives of Computational Methods in Engineering*, vol. 31, no. 1, pp. 125-146, 2024, doi: <https://doi.org/10.1007/s11831-023-09975-0>.

- [17] L. Serafino, "The no free lunch theorem: what are its main implications for the optimization practice?," in *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*: Springer, 2021, pp. 357-372.
- [18] A. H. Kashan, "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Applied Soft Computing*, vol. 16, pp. 171-200, 2014, doi: <https://doi.org/10.1016/j.asoc.2013.12.005>.
- [19] Y. Xu, Z. Cui, and J. Zeng, "Social emotional optimization algorithm for nonlinear constrained optimization problems," in *Swarm, Evolutionary, and Memetic Computing: First International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010, Chennai, India, December 16-18, 2010. Proceedings 1*, 2010: Springer, pp. 583-590, doi: https://doi.org/10.1007/978-3-642-17563-3_68.
- [20] D.-N. Truong and J.-S. Chou, "Metaheuristic algorithm inspired by enterprise development for global optimization and structural engineering problems with frequency constraints," *Engineering Structures*, vol. 318, p. 118679, 2024, doi: <https://doi.org/10.1016/j.engstruct.2024.118679>.
- [21] R. Zhong, J. Yu, C. Zhang, and M. Munetomo, "SRIME: a strengthened RIME with Latin hypercube sampling and embedded distance-based selection for engineering optimization problems," *Neural Computing and Applications*, vol. 36, no. 12, pp. 6721-6740, 2024, doi: <https://doi.org/10.1007/s00521-024-09424-4>.
- [22] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-73, 1992, doi: <https://www.jstor.org/stable/24939139>.
- [23] K. V. Price, "Differential evolution: a fast and simple numerical optimizer," in *Proceedings of North American Fuzzy Information Processing*, 1996: IEEE, pp. 524-527, doi: <https://doi.org/10.1109/NAFIPS.1996.534790>.
- [24] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702-713, 2008, doi: <https://doi.org/10.1109/TEVC.2008.919004>.
- [25] M. Valikhan Anaraki and S. Farzin, "The Pine Cone Optimization Algorithm (PCOA)," *Biomimetics*, vol. 9, no. 2, p. 91, 2024, doi: <https://doi.org/10.3390/biomimetics9020091>.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, vol. 4: ieeep, pp. 1942-1948, doi: <https://doi.org/10.1109/ICNN.1995.488968>.
- [27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [28] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017, doi: <https://doi.org/10.1016/j.advengsoft.2017.07.002>.
- [29] L. Xie, T. Han, H. Zhou, Z.-R. Zhang, B. Han, and A. Tang, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 9210050, 2021, doi: <https://doi.org/10.1155/2021/9210050>.
- [30] M. Abdel-Basset, R. Mohamed, M. Jameel, and M. Abouhawwash, "Spider wasp optimizer: A novel meta-heuristic optimization algorithm," *Artificial Intelligence Review*, vol. 56, no. 10, pp. 11675-11738, 2023, doi: <https://doi.org/10.1007/s10462-023-10446-y>.
- [31] M. N. Sholeh and L. Karlina, "An Innovative Metaheuristic Algorithm Inspired by Behavior of Birds-of-Paradise in Tropical Rainforests for Global Optimization," *Neural Computing and Applications*, 2025.

- [32] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009, doi: <https://doi.org/10.1016/j.ins.2009.03.004>.
- [33] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. Jawawi, "Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm," *Swarm and Evolutionary Computation*, vol. 26, pp. 8-22, 2016, doi: <https://doi.org/10.1016/j.swevo.2015.07.002>.
- [34] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, pp. 1531-1551, 2021, doi: <https://doi.org/10.1007/s10489-020-01893-z>.
- [35] R. Kundu, S. Chattopadhyay, S. Nag, M. A. Navarro, and D. Oliva, "Prism refraction search: A novel physics-based metaheuristic algorithm," *The Journal of Supercomputing*, vol. 80, no. 8, pp. 10746-10795, 2024, doi: <https://doi.org/10.1007/s11227-023-05790-3>.
- [36] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303-315, 2011, doi: <https://doi.org/10.1016/j.cad.2010.12.015>.
- [37] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, pp. 850-887, 2017, doi: <https://doi.org/10.1007/s10489-017-0903-6>.
- [38] M. Dehghani *et al.*, "A new "Doctor and Patient" optimization algorithm: An application to energy commitment problem," *Applied Sciences*, vol. 10, no. 17, p. 5791, 2020, doi: <https://doi.org/10.3390/app10175791>.
- [39] Q. Askari, I. Younas, and M. Saeed, "Political Optimizer: A novel socio-inspired metaheuristic for global optimization," *Knowledge-Based Systems*, vol. 195, p. 105709, 2020, doi: <https://doi.org/10.1016/j.knosys.2020.105709>.
- [40] M. Dehghani, E. Trojovská, and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process," *Scientific Reports*, vol. 12, no. 1, p. 9924, 2022, doi: <https://doi.org/10.1038/s41598-022-14225-7>.
- [41] A. T. Mozhdehi *et al.*, "Divine Religions Algorithm: a novel social-inspired metaheuristic algorithm for engineering and continuous optimization problems," *Cluster Computing*, vol. 28, no. 4, p. 253, 2025, doi: <https://doi.org/10.1007/s10586-024-04954-x>.
- [42] J.-S. Chou and C.-Y. Liu, "Pilgrimage walk optimization: Folk culture-inspired algorithm for identification of bridge deterioration," *Automation in Construction*, vol. 155, p. 105055, 2023, doi: <https://doi.org/10.1016/j.autcon.2023.105055>.
- [43] J.-S. Chou and A. Molla, "Arctic tern-optimized weighted feature regression system for predicting bridge scour depth," *Engineering Applications of Computational Fluid Mechanics*, vol. 18, no. 1, p. 2364745, 2024, doi: <https://doi.org/10.1080/19942060.2024.2364745>.
- [44] A. W. Mohamed, A. A. Hadi, A. K. Mohamed, and N. H. Awad, "Evaluating the performance of adaptive gainsharing knowledge based algorithm on CEC 2020 benchmark problems," in *2020 IEEE congress on evolutionary computation (CEC)*, 2020: IEEE, pp. 1-8, doi: <https://doi.org/10.1109/CEC48606.2020.9185901>.
- [45] M. F. Verde, N. A. Macmillan, and C. M. Rotello, "Measures of sensitivity based on a single hit rate and false alarm rate: The accuracy, precision, and robustness of", *A z*, and

- A'," *Perception & Psychophysics*, vol. 68, pp. 643-654, 2006, doi: <https://doi.org/10.3758/BF03208765>.
- [46] X. Li *et al.*, "A novel index of functional connectivity: phase lag based on Wilcoxon signed rank test," *Cognitive Neurodynamics*, vol. 15, pp. 621-636, 2021, doi: <https://doi.org/10.1007/s11571-020-09646-x>.
- [47] J. Liu and Y. Xu, "T-Friedman test: a new statistical test for multiple comparison with an adjustable conservativeness measure," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, p. 29, 2022, doi: <https://doi.org/10.1007/s44196-022-00083-8>.
- [48] M. Ravber, S.-H. Liu, M. Mernik, and M. Črepinšek, "Maximum number of generations as a stopping criterion considered harmful," *Applied Soft Computing*, vol. 128, p. 109478, 2022, doi: <https://doi.org/10.1016/j.asoc.2022.109478>.
- [49] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108-132, 2009, doi: <https://doi.org/10.1016/j.amc.2009.03.090>.
- [50] H. R. R. Zaman and F. S. Gharehchopogh, "An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems," *Engineering with Computers*, vol. 38, no. Suppl 4, pp. 2797-2831, 2022, doi: <https://doi.org/10.1007/s00366-021-01431-6>.
- [51] S. Gupta, H. Abderazek, B. S. Yıldız, A. R. Yildiz, S. Mirjalili, and S. M. Sait, "Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems," *Expert Systems with Applications*, vol. 183, p. 115351, 2021, doi: <https://doi.org/10.1016/j.eswa.2021.115351>.
- [52] S. Fu, K. Li, H. Huang, C. Ma, Q. Fan, and Y. Zhu, "Red-billed blue magpie optimizer: a novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems," *Artificial Intelligence Review*, vol. 57, no. 6, pp. 1-89, 2024, doi: <https://doi.org/10.1007/s10462-024-10716-3>.
- [53] A. Kaveh and S. Talatahari, "A charged system search with a fly to boundary method for discrete optimum design of truss structures," *Asian Journal of Civil Engineering (Building and Housing)*, 2010.
- [54] T. Dede, "Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures," *KSCE Journal of Civil Engineering*, vol. 18, pp. 1759-1767, 2014, doi: <https://doi.org/10.1007/s12205-014-0553-8>.
- [55] S. O. Degertekin, L. Lamberti, and I. B. Ugur, "Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm," *Applied Soft Computing*, vol. 79, pp. 363-390, 2019, doi: <https://doi.org/10.1016/j.asoc.2019.03.058>.
- [56] A. Sadollah, H. Eskandar, A. Bahreininejad, and J. H. Kim, "Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures," *Computers & Structures*, vol. 149, pp. 1-16, 2015, doi: <https://doi.org/10.1016/j.compstruc.2014.12.003>.
- [57] F. K. Jawad, M. Mahmood, D. Wang, A.-A. Osama, and A.-J. Anas, "Heuristic dragonfly algorithm for optimal design of truss structures with discrete variables," in *Structures*, 2021, vol. 29: Elsevier, pp. 843-862, doi: <https://doi.org/10.1016/j.istruc.2020.11.071>.
- [58] O. Altay, O. Cetindemir, and I. Aydogdu, "Size optimization of planar truss systems using the modified salp swarm algorithm," *Engineering Optimization*, vol. 56, no. 4, pp. 469-485, 2024, doi: <https://doi.org/10.1080/0305215X.2022.2160449>.