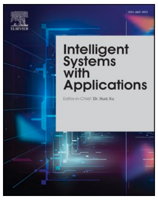
Intelligent Systems with Applications 25 (2025) 200478

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications



Intelligent Systems with Applications

A novel improved horned lizard optimization algorithm to identify optimal parameters of adaptive fuzzy logic MPPT for performance boosting of PEM fuel cell

Hegazy Rezk [a,b], Anas Bouaouda [c,*], Fatma A. Hashim [d,e]

[a] Department of Electrical Engineering, College of Engineering in Wadi Alddawasir, Prince Sattam bin Abdulaziz University, Saudi Arabia
[b] Faculty of Engineering, Minia University, Egypt
[c] Faculty of Science and Technology, Hassan II University of Casablanca, Mohammedia, Morocco
[d] Faculty of Engineering, Helwan University, Egypt
[e] Jadara University Research Center, Jadara University, Jordan

ARTICLE INFO

Keywords:
Horned lizard optimization algorithm
Fuzzy logic control
Fuel cell
Parameter identification
Metaheuristic algorithms

ABSTRACT

Horned Lizard Optimization Algorithm (HLOA) is a newly developed swarm-based metaheuristic technique that emulates the defensive behaviors of the horned lizard in nature. Like other algorithms, HLOA has certain limitations, including the tendency to become trapped in local optima due to a rapid loss of population diversity during the optimization process. This often results in premature convergence, particularly in complex optimization problems. To address these issues, this paper introduces an improved version of HLOA, named iHLOA, which incorporates two distinct strategies. First, the strengthened convergence strategy is utilized to improve the quality of individuals and accelerate the algorithm's convergence. Second, the mutation strategy is integrated to significantly boost population diversity, enhancing HLOA's ability to escape local minima. Various validation tests conducted on the CEC-2022 benchmark test demonstrate the effectiveness of the iHLOA algorithm in tackling global optimization challenges. Additionally, iHLOA was applied to determine the optimal gains for adaptive Fuzzy Logic Control (FLC) based MPPT to maximize energy harvested from the Proton Exchange Membrane Fuel Cell (PEMFC). The results demonstrate iHLOA's superiority over other algorithms, including the Seagull Optimization Algorithm (SOA), Black Widow Optimization Algorithm (BWOA), Sinh Cosh Optimizer (SCHO), Osprey Optimization Algorithm (OOA), Whale Optimization Algorithm (WOA), Greylag Goose Optimization (GGO), and the standard HLOA. iHLOA achieved the best performance with a value of 1.7755, followed by SCHO with 1.7806, while GGO recorded the worst performance at 1.8494. Additionally, iHLOA demonstrated superior stability with the lowest standard deviation (STD) of 0.0122, followed by SOA with 0.0193, while GGO had the highest STD of 0.1101. Furthermore, compared with the classical FLC-MPPT, the proposed FLC-MPPT based on iHLOA achieves faster tracking speeds and reduces oscillations around the MPP in a steady state.

1. Introduction

The core of optimization problems lies in identifying the optimal set of variables (parameters) that generate the best possible outcome (either minimum or maximum value) for a given design while complying with specific constraints (Rajwar, Deep, & Das, 2023). These problems are typically formulated as mathematical programming models. The need for practical solutions to optimization problems is widespread across nearly all scientific and engineering disciplines (Fatehi, Toloei, Niaki, & Zio, 2023). As the complexity of challenges in these fields grows, the demand for flexible optimization methods capable of handling their intricate nature becomes increasingly critical (Fathy, Bouaouda, & Hashim, 2024).

Metaheuristic algorithms have emerged as powerful tools for addressing complex optimization problems (Umbarkar, Sheth, Hong, & Jagdeo, 2024). Traditional methods often struggle with escaping suboptimal solutions (local optima) and can become trapped in these areas. In contrast, metaheuristics employ a variety of simulated behaviors to effectively explore the search space, thereby improving the likelihood of identifying the true optimal solution (global optimum) (Velasco,

[*] Corresponding author.
E-mail addresses: hr.hussien@psau.edu.sa (H. Rezk), anas.bouaouda-etu@etu.univh2c.ma (A. Bouaouda), fatma_hashim@h-eng.helwan.edu.eg (F.A. Hashim).

https://doi.org/10.1016/j.iswa.2025.200478
Received 25 August 2024; Received in revised form 17 December 2024; Accepted 5 January 2025
Available online 13 January 2025

Guerrero, & Hospitaler, 2024, Bouaouda & Sayouti, 2022). Moreover, metaheuristics are notably user-friendly, contributing to their widespread adoption in optimization tasks since their introduction.

Optimization algorithms have recently seen a surge in research activity, leading to the development of numerous innovative metaheuristic methodologies (Sharma & Raju, 2024). Notable examples include the Elk Herd Optimizer (EHO) (Al-Betar et al., 2024), Walrus optimizer (WO) (Han et al., 2024), Topology Aggregation Optimizer (TTAO) (Zhao, Zhang, Cai, & Yang, 2024), Growth Optimizer (GO) (Zhang et al., 2023), Arctic Puffin Optimization (APO) (Wang, Tian, Xu, & Zang, 2024), Puma optimizer (PO) (Abdollahzadeh et al., 2024), Pied Kingfisher Optimizer (PKO) (Bouaouda, Hashim, Sayouti, & Hussien, 2024), and Horned Lizard Optimization Algorithm (HLOA) (Peraza-Vázquez et al., 2024). These algorithms have demonstrated superior performance compared to established techniques in initial studies and are continually being refined, highlighting their significant potential (Yacoubi, Manita, Chhabra,& Korbaa, 2024). This expanding landscape of optimization algorithms highlights the ongoing pursuit of improved problem-solving capabilities. According to the “No Free Lunch” theorem (Wolpert & Macready, 1997), no single algorithm can be universally optimal for all problems. Consequently, current research focuses on developing algorithms with broader applicability and significantly improved optimization abilities across various problem domains.

The Proton Exchange Membrane Fuel Cell (PEMFC) is considered one of the most promising types of FCs thanks to its low operating temperature and extended lifespan (Ali, Al-Dhaifallah, Al-Gahtani, & Muranaka, 2023). The output power of a PEM fuel cell is regularly reliant on the cell temperature and membrane water content (Hai, Alazzawi, Zhou, & Farajian, 2023). Therefore, maximum power point tracking systems are highly integrated with the PEMFC to ensure maximum power harvesting. A new MPPT approach for tracking the MPP of PEM fuel cells using ANFIS with a modified manta ray foraging algorithm is proposed by Ali et al. (Ali, Al-Dhaifallah, Al-Gahtani, & Muranaka, 2023). The drawback of this method is that it is required to measure the membrane water content (MWC) and the temperature in addition to the voltage and current. A new method using an improved Fluid Search Optimization Algorithm and FLC to extract the MPP of PEMFC is suggested by Hai, Alazzawi, Zhou, & Farajian (2023). Different values of cell temperature and membrane water content are considered. A non-iterative method for MPPT of PEMFC using resistance estimation is presented by Bankupalli et al. (2019). The findings were compared with the incremental resistance technique. The drawback of the suggested method is that extra sensors are required, including two voltage sensors and two current sensors. Fathy, Abdelkareem, Olabi, & Rezk (2021) examined a new strategy using the Salp Swarm Algorithm (SSA) to track the MPP of PEMFC. A mathematical relation is presented to calculate the PEMFC voltage value matching the MPP. The error between the computed voltage at MPP and the actual PEMFC voltage is fed to a PID to tune the converter duty cycle to boost the PEMFC power. The ideal gains values of PID are determined using SSA, and the findings were compared with INC, FLC, GWO, ALO, and MPA (Fathy, Abdelkareem, Olabi, & Rezk, 2021). The drawback of the proposed system is that it needs three sensors to measure the temperature, MWC, and cell voltage.

According to the literature, several challenges remain in improving the MPPT performance of PEMFC systems, including enhancing tracking speeds, optimizing design parameters, reducing steady-state oscillations, lowering sensor costs, and simplifying implementation. Additionally, the stochastic nature of optimization techniques in single-run designs necessitates statistical analysis for meaningful comparisons. The choice of control method, particularly FLC, plays a critical role in MPPT performance, as it offers design flexibility for performance optimization. In addressing these challenges, this study proposes an enhanced MPPT controller design for PEMFCs, incorporating FLC systems to improve overall efficiency.

However, while FLCs offer significant benefits, their design remains complex due to the lack of standardized methods for defining fuzzy control rules and fine-tuning membership function (MF) parameters. The design process is often formulated as an optimization problem to address these challenges, typically solved using metaheuristic algorithms (Hai, Alazzawi, Zhou, & Farajian, 2023). These algorithms automatically adjust both input and output MFs, leading to more effective tuning. They also optimize the scaling factors (SFs) of MFs and FLC parameters, thus reducing the number of variables that need to be optimized. Focusing on SF adjustments instead of directly modifying the fuzzy set ranges minimizes computational demands while improving tracking speed. A key advantage of these algorithms is their ability to automatically regulate the shape and range of triangular MFs, enhancing the adaptability and performance of FLCs.

The Horned Lizard Optimization Algorithm (HLOA) is a recently introduced method by Peraza-Vázquez et al. (2024)., inspired by the defensive behaviors of horned lizards in nature to seek optimal solutions. Since its introduction, HLOA has attracted considerable attention due to its minimal parameters and straightforward design. However, its application to complex, real-world scenarios has revealed limitations, particularly its tendency to become trapped in local optima. To address this, an improved version called iHLOA has been developed, incorporating strengthened convergence and mutation strategies. These improvements are intended to help HLOA effectively escape local optima, enhance its exploration capabilities, and more reliably guide the algorithm toward the global optimal solution.

The main contributions of the paper are as follows.

(a) An improved version of the HLOA algorithm, referred to as iHLOA, is proposed.
(b) The improved iHLOA incorporates two distinct strategies. **First**, the strengthened convergence strategy is used to improve the quality of individuals and accelerate the algorithm’s convergence. **Second**, the mutation strategy is integrated to boost population diversity significantly, enhancing the ability of HLOA to escape local minima
(c) The proposed approach was evaluated using the CEC-2022 benchmark suite and compared against other widely and popular techniques.
(d) The iHLOA was applied for the first time to determine the optimal gains of adaptive fuzzy logic control-based maximum power point tracking.
(e) The suggested FLC-MPPT-based iHLOA presented fast tracking speed and overcame the oscillations around MPP in a steady state.

The structure of the paper is as follows: Section 2 reviews related works. Section 3 overviews the horned lizard optimization algorithm. Section 4 introduces the proposed improvements to HLOA. Section 5 presents the experimental results and analysis of the proposed approach across various benchmark functions. Section 6 examines the use of the improved HLOA (iHLOA) in optimizing FLC-based MPPT. Section 7 discusses the merits and demerits of the proposed method. Lastly, Section 8 summarizes the conclusions and key findings of the paper.

2. Related works

Numerous MPPT techniques have been developed to control the output of PEMFCs, with the primary goal of maintaining continuous operation at the MPP. Traditional approaches, including Incremental Conductance (INC) (Harrag & Messalti, 2017), Perturb and observe (P&O) (Benyahia et al., 2014), Incremental resistance (INR) (Rezk & Fathy, 2020), and Sliding Mode Control (SMS) (Derbeli, Barambones, Ramos-Hernanz, & Sbita, 2019), are commonly employed. These methods focus on control variables. Although these techniques are relatively straightforward to implement, they often experience power oscillations around the MPP. Decreasing the step size can alleviate these oscillations and reduce tracking speed, limiting adaptability to rapidly changing conditions. The main challenge remains to find the right balance between reducing power fluctuations and ensuring rapid, precise tracking.

To overcome the shortcomings of traditional MPPT methods in PEMFC systems, intelligent control algorithms are increasingly utilized. Fuzzy Logic Control (FLC)-based MPPT strategies, as discussed in Ref. (Derbeli, Sbita, Farhat, & Barambones, 2017), leverage the adaptability and flexibility of FLC systems to enhance MPPT effectiveness for PEMFCs. FLC offers advantages such as fast response, operational flexibility, and independence from precise mathematical models. However, it depends on specialized knowledge to establish fuzzy inference rules, which means its effectiveness depends on the user's expertise. Additionally, FLC may require significant memory resources, leading to higher overall expenses for the control system.

Ref. Reddy and Sudhakar (2018) presents MPPT controllers based on Artificial Neural Networks (ANN) for PEMFC systems. ANN controllers can effectively approximate the voltage and current at the MPP across varying cell temperatures. However, they require extensive training datasets, which can increase computational demands. Insufficient training data can significantly reduce the accuracy of MPP tracking. In Ref. Reddy and Sudhakar (2019), an Adaptive Neuro-Fuzzy Inference System (ANFIS)-based MPPT controller was developed for PEMFC-powered electric vehicles. By combining the strengths of FLC and ANN, ANFIS controllers were tested across a range of cell temperatures. Despite their advantages, ANFIS controllers rely on substantial training data, impacting system efficiency.

Beyond the controllers mentioned earlier, metaheuristic algorithms have been used to optimize the output of PEMFC systems. These algorithms, recognized for their ability to quickly and accurately identify the MPP, have garnered significant interest in optimizing MPPT for various renewable energy systems. Notable examples include Ref. Ahmadi, Abdi, and Kakavand (2017), which introduced an enhanced PID control approach utilizing Particle Swarm Optimization (PSO) to improve the stability and accuracy of the system. Compared to the P&O method and sliding mode algorithm, PSO achieves high precision and reduces power fluctuations under varying temperature and membrane water content conditions. However, PSO is prone to premature convergence, parameter sensitivity, and can be computationally intensive. On the other hand, the Genetic Algorithm (GA) has been proposed to optimize MPPT via ANFIS for fuel cell systems (Savrun and İnci, 2021). This method reduces computational load, making it suitable for real-time applications. A performance comparison with ANFIS and PI controllers demonstrated its effectiveness in terms of power extraction and efficiency. However, GA-based optimization faces convergence issues in complex environments.

In Ref. Nasiri Avanaki and Sarvi (2016), an MPPT method for PEMFCs based on the Water Cycle Algorithm (WCA)-PID combination was proposed, where WCA identified the voltage at maximum power, and the PID controller adjusted the duty cycle. While WCA outperformed P&O and voltage/current-based methods, it faces challenges during the exploitation phase. Ref. Rana, Kumar, Sehgal, and George (2019) introduced a feedback-based control system integrating the GWO-tuned PID controller for efficient MPPT. Despite improved tracking efficiency, it remains prone to local optima challenges, particularly in fluctuating conditions. Ref. Percin and Caliskan (2023) proposed an MPPT strategy for PEMFCs using the Whale Optimization Algorithm (WOA). Although the method is noted for its straightforward structure and ease of implementation, it tends to converge more slowly in complex environments and is highly sensitive to parameter tuning.

Table 1
Summary of various MPPT controllers for PEMFC systems.

Ref.	Method	*Abbrev.*	Control variable	Type of Converter	Strengths	Weaknesses
(Benyahia et al., 2014)	Perturb and observe	P&O	Duty cycle	Boost converter	It is easy to implement and requires minimal computational resources.	It may struggle with fast-changing conditions and can cause oscillations around the MPP.
(Harrag & Messalti, 2017)	Incremental Conductance	INC	Duty cycle	Boost converter	It is more accurate than P&O under rapidly changing conditions and less oscillatory.	Computationally more complex than P&O and requires derivative calculations
(Rezk & Fathy, 2020)	Incremental Resistance	INR	Duty cycle	Boost converter	Accurate for MPPT in specific conditions, capable of steady-state operation.	Limited tracking during rapid variations in operating conditions.
(Derbeli, Barambones, Ramos-Hernanz, & Sbita, 2019)	Sliding Mode Control	SMC	Duty cycle	Boost converter	Robust to parameter variations and disturbances, making it suitable for fuel cells.	It may suffer from chattering and requires careful design.
(Derbeli, Sbita, Farhat, & Barambones, 2017)	Fuzzy Logic Control	FLC	Duty cycle	Boost converter	Effective in handling the non-linearity of FCs and can adapt to changes in operating conditions.	Designing the rule base and membership functions can be complex and requires expert knowledge.
(Reddy & Sudhakar, 2018)	Artificial Neural Network	ANN	Duty cycle	High gain boost	High accuracy and adaptability to complex, non-linear relationships in PEMFC.	It requires a large amount of data for training and can be computationally demanding.
(Reddy & Sudhakar, 2019)	Neuro-Fuzzy Inference System	ANFIS	Duty cycle	High gain boost	Adaptable with high convergence and accuracy without requiring expert knowledge.	It requires extensive data and is computationally expensive.
(Ahmadi, Abdi, & Kakavand, 2017)	Particle Swarm Optimization	PSO	Duty cycle	Boost converter	High tracking accuracy and efficient performance.	Has a poor rate of convergence in some cases.
(Savrun & İnci, 2021)	Genetic Algorithm	GA	Duty cycle	Boost converter	Effective in handling complex optimization tasks without requiring derivative information	Optimizing it is computationally complex and time-intensive.
(Nasiri Avanaki & Sarvi, 2016)	Water Cycle Algorithm	WCA	Duty cycle	Boost converter	Offers high convergence speed and robust performance under certain conditions.	Parameter sensitivity can result in inaccurate MPPT tracking, as improperly tuned parameters may adversely affect performance.
(Fathy, Abdelkareem, Olabi, & Rezk, 2021)	Salp Swarm Algorithm	SSA	Duty cycle	Boost converter	Suitable for various environmental conditions.	Slow convergence in some scenarios may lead to local minima under complex conditions.
(Rana, Kumar, Sehgal, & George, 2019)	Grey Wolf Optimizer	GWO	Duty cycle	Boost converter	Effective in handling complex search spaces.	Can experience premature convergence to suboptimal solutions.
(Percin & Caliskan, 2023)	Whale Optimization Algorithm	WOA	Duty cycle	Boost converter	It has a simple structure and is easy to implement.	It converges slower in complex environments and is sensitive to parameter adjustments.

While these algorithms are effective in MPP tracking, they face limitations in real-time applications. Metaheuristic algorithms typically require precise parameter tuning and may suffer from premature convergence, leading to suboptimal solutions. Moreover, their performance can degrade under rapidly changing environmental conditions, compromising consistent tracking accuracy. Table 1 summarizes different MPPT control strategies for PEMFC systems, highlighting their drawbacks.

3. Overview of the horned lizard optimization algorithm

The Phrynosoma, commonly known as the horned lizard, is a reptile found in northeastern Mexico and the south-central US's dry (arid and semi-arid) environments (Cooper & Sherbrooke, 2010). These lizards are adapted to their challenging habitat (see Fig. 1). They are primarily myrmecophagous, meaning they predominantly feed on ants, employing a sit-and-wait predatory strategy that relies on camouflage (cryptic coloration) to ambush their prey. Their distinctive morphology, which includes cranial horns, further supports their predatory lifestyle (Middendorf & Sherbrooke, 1992). A unique antipredator adaptation of the horned lizard is the ability to squirt blood from its eyes. These adaptations, along with their documented low annual survival rates (ranging from 8.9 % to 54 %), underscore the intense predation pressures these lizards face from a range of vertebrate predators, such as snakes, other predatory lizards, felids, canids, rodents, and birds (Sherbrooke, Aguilar-Morales, & Van Devender, 2022).

Peraza-Vázquez et al. introduced the Horned Lizard Optimization Algorithm (HLOA), drawing inspiration from the defensive behaviors of the horned lizard (Peraza-Vázquez et al., 2024). This algorithm converts these behaviors into mathematical models to guide the optimization process. One key behavior integrated into HLOA is crypsis, where the lizard adjusts its coloration to become translucent, aiding in evading predators. HLOA incorporates the concept of α-Melanophore Hormone Stimulation (α-MHS) to simulate adaptive skin lightening or darkening based on thermal regulation needs. Moreover, the algorithm mathematically represents the lizard's evasive movement strategy when encountering threats. Finally, HLOA models the horned lizard's unique defense mechanism of blood squirting, treating it as a projectile motion problem within its optimization framework (Peraza-Vázquez et al., 2024).

The HLOA was devised into six stages, which are described as follows.

3.1. Initialization

HLOA begins the optimization process by using a random initialization technique to create a population of candidate solutions within the problem's search space using the equation below.

$$X = LB + rand(UB - LB) \tag{1}$$

where UB and LB represent the upper and lower boundaries of the problem variables, and $rand$ is a uniform random vector ranging from 0 to 1.

3.2. Crypsis behavior strategy

Many organisms in the animal kingdom have evolved a remarkable ability known as crypsis, which involves camouflaging themselves by mimicking elements of their surroundings, such as color and texture. Some species even take it a step further by becoming translucent, further obscuring their presence. This adaptive behavior provides a significant advantage: it allows organisms to evade detection by predators and prey, ultimately enhancing their chances of survival. Inspired by this fascinating phenomenon, the following equation utilizes the principles of color theory to mathematically model the crypsis technique employed by the horned lizard (Peraza-Vázquez et al., 2024).

$$\vec{X}_i(t+1) = \vec{X}_{best}(t) + \left(\partial - \frac{\partial \cdot t}{T}\right) \times \left[c_1\left(\sin\left(\vec{X}_{r_1}(t)\right) - \cos\left(\vec{X}_{r_2}(t)\right)\right) - (-1)^{\sigma} c_2\left(\cos\left(\overrightarrow{X_{r_3}(t)}\right) - \sin\left(\vec{X}_{r_4}(t)\right)\right)\right] \tag{2}$$

where $\vec{X}_i(t+1)$ denotes the new position of the search agent within the solution search space for the generation $t+1$, $\vec{X}_{best}(t)$ represents the position of the best search agent identified so far in the current iteration, r_1, r_2, r_3, and r_4 represent integer values chosen randomly between 1 and the total number of search agents (denoted by N), which are distinct from each other ($r_1 \neq r_2 \neq r_3 \neq r_4$), T represents the maximum number of iterations, t is the current iteration number, and ∂ is a constant value equal to 2. Additionally, c_1 and c_2 are random numbers selected from a list of the normalized color palettes in which $c_1 \neq c_2$. Finally, σ is assigned a binary value based on a random number generation process. If the randomly generated value, denoted by rand, is less or equal to 0.5, σ is set to 0. Conversely, if rand is greater than 0.5, σ takes on a value of 1.

3.3. Skin darkening or lightening strategy

The horned lizard exhibits an impressive thermoregulatory adaptation by dynamically adjusting its skin coloration. It can darken or lighten its skin to control solar thermal energy absorption. This strategy is based on the fundamental concept of energy conservation, which applies to thermal and light energy. This principle highlights the crucial relationship between an organism's coloration and thermal state. Lighter colors reflect more solar radiation, reducing heat gain, while darker colors increase heat absorption by capturing more light energy. The mathematical representation of this skin-lightening strategy can be expressed as follows (Peraza-Vázquez et al., 2024):

$$\begin{aligned}\vec{X}_{worst}(t) = \vec{X}_{best}(t) + \frac{1}{2}Light_1 \sin\left(\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)\right) \\ - (-1)^{\sigma}\frac{1}{2}Light_2 \sin\left(\vec{X}_{r_3}(t) - \vec{X}_{r_4}(t)\right)\end{aligned} \tag{3}$$

where $\vec{X}_{best}(t)$ and $\vec{X}_{worst}(t)$ denote the best and worst performing search agents identified so far, respectively. $Light_1$ and $Light_2$ are random numbers generated within the range defined by $Lightening_1$ (value 0) and $Lightening_2$ (value 0.4046661). On the other hand, the mathematical representation of this skin-darkening strategy can be formulated as below (Peraza-Vázquez et al., 2024):

$$\begin{aligned}\vec{X}_{worst}(t) = \vec{X}_{best}(t) + \frac{1}{2}Dark_1 \sin\left(\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)\right) \\ - (-1)^{\sigma}\frac{1}{2}Dark_2 \sin\left(\vec{X}_{r_3}(t) - \vec{X}_{r_4}(t)\right)\end{aligned} \tag{4}$$

where $\vec{X}_{best}(t)$ and $\vec{X}_{worst}(t)$ denote the best and worst performing search agents identified so far, respectively. $Dark_1$ and $Dark_2$ are random numbers generated within the range defined by $Darkening_1$ (0.5440510 value) and $Darkening_2$ (1 value). Furthermore, within both equations, r_1, r_2, r_3, and r_4 represent integer values chosen randomly between 1 and

a) Adaptive coloration

b) Body expand

c) Fast movement

d) Squirt blood from their eyes

Fig. 1. Horned Lizard defense behaviors in nature.[1].

the total number of search agents (denoted by N), which are distinct from each other ($r_1 \neq r_2 \neq r_3 \neq r_4$).

Note that the worst search agent in iteration t is replaced by a new one generated through the skin-lightening or skin-darkening strategy.

3.4. Blood squirting strategy

The horned lizard exhibits a unique antipredator adaptation by expelling blood from its eyes. This hemoptysis defense mechanism can be mathematically modeled as projectile motion. To formulate the equations of motion, we decompose the blood trajectory into its horizontal (X-axis) and vertical (Y-axis) components.

In the horizontal direction, the blood shot follows a uniform linear movement, described by the following equation of motion (Peraza-Vázquez et al., 2024):

$$\vec{v} = \vec{v}_0 + \int_0^t \vec{g}\,dt = \vec{v}_0 + \vec{g}\,t \tag{5}$$

In the vertical direction, the blood shot follows a uniformly accelerated rectilinear motion, described as follows (Peraza-Vázquez et al., 2024):

$$\vec{r} = \vec{r_0} + \int_0^t (\vec{v_o} + \vec{g}\ t)dt = \vec{r_0} + \vec{v_o}\ t + \frac{1}{2}\vec{g}\ t^2 \tag{6}$$

In which,

$$\vec{r}_0 = \vec{0} \tag{7}$$

The equations below formally express the vector equations, position, and velocity (Peraza-Vázquez et al., 2024).

$$\vec{v}_0 = v_0\cos(\alpha)t\vec{j} + \left(v_0\sin(\alpha)t - \frac{1}{2}gt^2\right)\vec{k} \tag{8}$$

$$\vec{v} = \vec{r} = (v_0\cos(\alpha))\vec{j} + (v_0\sin(\alpha) - gt)\vec{k} \tag{9}$$

Lastly, the trajectory can be formulated as follows (Peraza-Vázquez et al., 2024):

$$\vec{X}_i(t+1) = \left[v_o\cos\left(\alpha\frac{t}{T}\right) + \varepsilon\right]\vec{X}_{best}(t) + \left[v_o\sin\left(\alpha - \frac{\alpha t}{T}\right) - g + \varepsilon\right]\vec{X}_i(t) \tag{10}$$

where g denotes the Earth's gravity (0.009807 km/s^2), ε is set to 1E-6, v_o is set to 1 second, t denotes the current iteration, T represents the maximum number of iterations (generations), $\vec{X}_{best}(t)$ represents the best search agent found at iteration t, and $\vec{X}_i(t+1)$ denotes the new position of the search agent within the solution search space for the generation $t+1$.

[1] Pictures obtained from https://pixabay.com/ are copyright-free.

3.5. Move-to-escape strategy

The horned lizard's survival strategy includes a remarkable evasion tactic. When threatened, it rapidly and unpredictably changes direction, confusing predators and making it difficult to anticipate its movements. HLOA mathematically represents this strategy through an equation incorporating local and global movements. The local exploration component, denoted by $walk\left(\frac{1}{2} - \varepsilon\right)\overrightarrow{X}_i(t)$, which guides the search agent's movement near its current position ($\overrightarrow{X}_i(t)$). Additionally, $\overrightarrow{X}_{best}(t)$ introduces a global movement aspect, influencing the search agent's trajectory towards promising regions within the solution space. Therefore, this strategy is expressed as follows (Peraza-Vázquez et al., 2024):

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_{best}(t) + walk\left(\frac{1}{2} - \varepsilon\right)\overrightarrow{X}_i(t) \tag{11}$$

where $\overrightarrow{X}_{best}(t)$ represents the best search agent found at iteration *t*, and $\overrightarrow{X}_i(t+1)$ denotes the new position of the search agent within the solution search space for the generation $t+1$, *walk* denotes a random value generated between -1 and 1, while ε represents a random variable sampled from a standard Cauchy distribution with a mean of 0 and a standard deviation (σ) of 1.

3.6. σ -Melanophore stimulating hormone rate strategy

The horned lizard exhibits a remarkable thermoregulatory adaptation. It can dynamically adjust its skin coloration, lightening, or darkening to manage its solar thermal energy absorption. This rapid color-changing process is primarily driven by the influence of temperature on α-melanophore stimulating hormone (α-MSH) concentrations. Consequently, the mathematical representation for a horned lizard's α-melanophore rate is expressed in the following equation (Peraza-Vázquez et al., 2024):

$$melanophore(i) = \frac{Fitness_{max} - Fitness(i)}{Fitness_{max} - Fitness_{min}} \tag{12}$$

where $Fitness(i)$ represents the objective function value (fitness) currently associated with the *i*-th search agent in the population. The terms $Fitness_{max}$ and $Fitness_{min}$ denote the worst and best fitness values observed within the current generation (t), respectively.

Following the calculation in the previous equation, the $melanophore(i)$ is normalized to a range between 0 and 1. This normalization ensures that the values remain within a valid range for further calculations. Subsequently, if the α-melanophore rate (represented by a value in $melanophore(i)$) falls below a threshold of 0.3, indicating a low melanophore stimulation level, the corresponding search agent is updated using the equation presented below (Peraza-Vázquez et al., 2024).

$$\overrightarrow{X}_i(t) = \overrightarrow{X}_{best}(t) + \frac{1}{2}\left[\overrightarrow{X}_{r_1}(t) - (-1)^{\sigma}\overrightarrow{X}_{r_2}(t)\right] \tag{13}$$

where $\overrightarrow{X}_i(t)$ denotes the current position of the search agent at iteration t, $\overrightarrow{X}_{best}(t)$ represents the position of the best search agent identified so far in the current iteration (t), while r_1 and r_2 are two randomly chosen integer values, each selected from the range of 1 to the total number of search agents, which are distinct from each other ($r_1 \neq r_2$). Additionally, σ is assigned a binary value based on a random number generation process. If the randomly generated value, denoted by rand, is less or equal to 0.5, σ is set to 0. Conversely, if rand is greater than 0.5, σ takes on a value of 1.

The pseudo-code of the HLOA algorithm is shown in Algorithm 1.

Algorithm 1
The HLOA algorithm.

```
Input: Upper and lower bounds of variables ($X_{max}$ and $X_{min}$),
problem dimension ($Dim$), the maximum number of iterations ($T$),
and number of population ($N$).
Output: the best solution
Begin
  Randomly initialize the population by Eq. (1).
  Calculate the fitness of all individuals.
  Initialize iteration count: $t = 1$.
  While ($t <$T)
    For $i$ =1 to $N$ do
      If ($0.5 < rand$) then
        Update the position by Eq. (2).
      Else
        If ($t =$ odd number) then
          Update the position by Eq. (10).
        Else
          Update the position by Eq. (11).
        End if
      End if
      Generate a random binary value $\sigma$ (0 or 1)
      If ($\sigma$ is equal to 0) then
        Generate randomly $Light_1$ and $Light_2$
        Replace the worst agent by Eq. (3).
      Else
        Generate randomly $Dark_1$ and $Dark_2$
        Replace the worst agent by Eq. (4).
      End if
      Calculate $melanophore(i)$ by Eq. (12).
      If ($melanophore(i) \leq 0.3$) then
        Update the position by Eq. (13).
      End if
      Check if the new position is within boundaries.
      Evaluate the current fitness.
    End for
    $t = t + 1$
  End while
End
```

4. Proposed improved HLOA

While the HLOA algorithm tends to converge towards optimal solutions, it faces challenges when applied to complex optimization problems. These challenges include susceptibility to getting trapped in suboptimal solutions (local optima) and inefficient search space exploration, which can lead to neglecting potentially optimal regions. To address these limitations, this work proposes an improved version of HLOA named iHLOA. iHLOA incorporates two key enhancements. Firstly, it introduces the strengthened convergence strategy to accelerate the search process and guide the population towards the optimal solution more efficiently. Secondly, the mutation strategy enhances the algorithm's ability to escape from local optima and explore the search space more effectively. These modifications aim to overcome the limitations of the original HLOA, as detailed in the following subsections.

4.1. Strengthened convergence strategy

In the HLOA algorithm, the population is guided by the optimal individual. However, due to the lack of prior knowledge, it is challenging to determine if this optimal individual is in the globally optimal position. If the optimal individual is in a locally optimal position, the population will converge to local optima, leading to suboptimal solutions. To address this issue, the strengthened convergence strategy is introduced (Wang et al., 2024). This strategy enables individuals to progressively move towards the global optimal solution by learning from those with better fitness values within the archive and distancing themselves from individuals with the worst fitness values in the population. This approach improves the quality of the individuals and accelerates HLOA's convergence. The strengthened convergence strategy is mathematically represented as follows (Wang et al., 2024).

$$\begin{cases} \overrightarrow{X}_i(t+1) = \dfrac{\overrightarrow{X}_{r_1}(t) + \overrightarrow{X}_{r_2}(t) + \overrightarrow{X}_{r_3}(t)}{3}, & \dfrac{\mathrm{t}}{\mathrm{T}} < 0.4 \\ \overrightarrow{X}_i(t+1) = \dfrac{\overrightarrow{X}_{r_1}(t) + \overrightarrow{X}_{r_2}(t)}{2} - \mathrm{Distance(i)}, & \dfrac{\mathrm{t}}{\mathrm{T}} \geq 0.4 \end{cases} \tag{14}$$

where $\overrightarrow{X}_i(t+1)$ denotes the new position of the search agent within the solution search space for the generation $t+1$, r_1, r_2, r_3, and r_4 represent integer values chosen randomly between 1 and the total number of search agents, which are distinct from each other ($r_1 \neq r_2 \neq r_3 \neq r_4$), T represents the maximum number of iterations, and t is the current iteration number. Furthermore, Distance(i) represents the distance between the current i-th individual and the position of the worst individual in the population, which is expressed below (Wang et al., 2024):

$$\mathrm{Distance(i)} = 0.9 \times \mathrm{EC} \times \left(\overrightarrow{X}_{worst}(t) - \left|\overrightarrow{X}_i(t)\right|\right) \tag{15}$$

where $\overrightarrow{X}_i(t)$ denotes the current position of the search agent at iteration t, $\overrightarrow{X}_{worst}(t)$ represents the position of the worst search agent identified so far in the current iteration, while EC is an adaptive elite coefficient of exploration formulated as follows (Wang et al., 2024):

$$EC = \left|1 - \frac{\mathrm{t}}{\mathrm{T}}^{\frac{1}{12} - \frac{5*\tan\,(pi*(\mathrm{rand}-0.5))}{3*\mathrm{T}}}\right| \tag{16}$$

where $rand$ is a uniform random vector ranging from 0 to 1, t is the current iteration number, while T represents the maximum number of iterations.

4.2. Mutation strategy

HLOA is recognized for its strong exploitation capabilities and simplicity, but it can suffer from reduced population diversity and premature convergence to local optima in later iterations. The introduced mutation strategy addresses this issue by combining states from multiple individuals to generate new solutions, thereby preserving diversity and facilitating exploring a broader solution space. When integrated with other metaheuristic algorithms, this approach leverages their global search strengths, mitigating the risk of local optima and enhancing the algorithm's performance in solving complex optimization problems (Chen, Ouyang, Li, & Zou, 2024). The updated formula for HLOA, incorporating the mutation strategy, is presented below.

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_i(t) + \beta \times rand \times \left(\overrightarrow{X}_{r_1}(t) - \overrightarrow{X}_i(t)\right) \tag{17}$$

where $\overrightarrow{X}_i(t+1)$ denotes the new position of the search agent within the solution search space for the generation $t+1$, $\overrightarrow{X}_i(t)$ denotes the current position of the search agent at iteration t, r_1 represents an integer value chosen randomly between 1 and the total number of search agents, $rand$ is a uniform random vector ranging from 0 to 1, and β is an updating coefficient that determines the step size for updating the individual.

A detailed description of the search process for the developed iHLOA is provided in Algorithm 2.

4.3. Computational complexity of the proposed iHLOA

This section analyzes the time and space complexity of the proposed iHLOA to assess the computational cost associated with the incorporated improvements.

4.3.1. Time complexity

For iHLOA, with N solutions, T maximum iterations, and D dimensions, the time complexity can be divided into distinct phases. The initialization phase, which occurs once it requires $O(N \times D)$. The complexity of the fitness function depends on the specific problem. On the other hand, the position update phase, which includes strategies such as strengthened convergence, mutation, skin lightening/darkening, blood squirting strategy, move-to-escape strategy, and σ-Melanophore stimulating hormone rate, contributes $O(T \times N \times D)$. Therefore, the overall computational complexity of iHLOA is as follows:

$$O(iHLOA) = O(N \times D) + O(T \times N \times D) \tag{18}$$

$$O(iHLOA) = O(N \times D + T \times N \times D) \tag{19}$$

$$O(iHLOA) \approx O(T \times N \times D) \tag{20}$$

As a result, iHLOA maintains the same time complexity ($O(T \times N \times D)$) as the original HLOA, but it significantly outperforms the original algorithm in terms of efficiency and solution quality.

Algorithm 2
The proposed iHLOA algorithm.

```
Input: Upper and lower bounds of variables ($X_{max}$ and $X_{min}$),
problem dimension ($Dim$), the maximum number of iterations ($T$),
and number of population ($N$).
Output: the best solution
Begin
  Randomly initialize the population by Eq. (1).
  Calculate the fitness of all individuals.
  Initialize iteration count: $t = 1$.
  While ($t$ <T)
    For $i$ =1 to $N$ do
      If ($rand < 0.3$) then
        Calculate EC by Eq. (16).
        Update the position by Eq. (14).
      Elseif ($rand < 0.8$) then
        Update the position by Eq. (17).
      Else
        If ($t$ = odd number) then
          Update the position by Eq. (10).
      Else
        Update the position by Eq. (11).
      End if
    End if
    Generate a random binary value $\sigma$ (0 or 1)
      If ($\sigma$ is equal to 0) then
        Generate randomly $Light_1$ and $Light_2$
        Replace the worst agent by Eq. (3).
      Else
        Generate randomly $Dark_1$ and $Dark_2$
        Replace the worst agent by Eq. (4).
      End if
      Calculate $melanophore(i)$ by Eq. (12).
      If ($melanophore(i) \leq 0.3$) then
        Update the position by Eq. (13).
      End if
      Check if the new position is within boundaries.
      Evaluate the current fitness.
    End for
    $t = t + 1$
  End while
End
```

4.3.2. Space complexity

In the case of iHLOA, memory allocation depends on the number of search agents and the dimension size determined during initialization. As a result, the space complexity of iHLOA is $O(N \times D)$.

The following sections highlight the advantages of iHLOA in addressing various optimization problems, using multiple benchmark functions, and a real-world engineering application.

5. Experimental results and analysis

This section provides an in-depth evaluation of iHLOA's optimization

Table 2
Details of the benchmark functions.

F	Name	Dim	Range	f_i^*
1	Shifted and full Rotated Zakharov Function	20	[−100,100]	300
2	Shifted and full Rotated Rosenbrock's Function	20	[−100,100]	400
3	Shifted and full Rotated Expanded Schaffer's	20	[−100,100]	600
4	Shifted and full Rotated Non-Continuous Rastrigin's	20	[−100,100]	800
5	Shifted and full Rotated Levy Function	20	[−100,100]	900
6	Hybrid Function 1 (N = 3)	20	[−100,100]	1800
7	Hybrid Function 2 (N = 6)	20	[−100,100]	2000
8	Hybrid Function 3 (N = 5)	20	[−100,100]	2200
9	Composition Function 1 (N = 5)	20	[−100,100]	2300
10	Composition Function 2 (N = 4)	20	[−100,100]	2400
11	Composition Function 3 (N = 5)	20	[−100,100]	2600
12	Composition Function 4 (N = 6)	20	[−100,100]	2700

performance through various numerical experiments using established test functions. To ensure consistency, all algorithms are implemented in MATLAB. The source code for the comparison algorithms is accessible from the original authors. Importantly, all algorithms are assessed under identical conditions within the same test category.

5.1. Benchmark description

This section utilizes benchmark functions from the CEC-2022 suite (Ahrari et al., 2022) to rigorously evaluate algorithm performance across diverse optimization landscapes. Table 2 provides comprehensive details about these functions, including their names, value ranges, dimensions (*Dim*), and optimal values (f_i^*). The function set includes unimodal (F1), multimodal (F2-F4), hybrid (F5-F8), and composite (F9-F12) functions, facilitating a thorough assessment of each algorithm's performance across different optimization scenarios. Each function is denoted by 'F' followed by a number (F1, F2, ..., F12). Unimodal functions, characterized by a single global optimum, are used to test an algorithm's exploitation capability, i.e., its ability to find the best solution. Multimodal functions, which have multiple optima (one global and others local), evaluate an algorithm's exploration ability or its capacity to navigate various options to find the best solution. Hybrid and composite functions simulate real-world complexities, assessing an algorithm's ability to handle diverse and intricate problem structures. Fig. 2 presents 3D visualizations of a randomly selected subset of the CEC-2022 test functions, offering a visual representation of the optimization landscapes.

5.2. Parameter setting

This study compares the performance of the proposed iHLOA with seven established optimization algorithms including the Seagull Optimization Algorithm (SOA) (Dhiman & Kumar, 2019), Black Widow Optimization Algorithm (BWOA) (Hayyolalam & Kazem, 2020), Sinh Cosh Optimizer (SCHO) (Bai et al., 2023), Osprey Optimization Algorithm (OOA) (Dehghani and Trojovský, 2023), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016), Greylag Goose Optimization

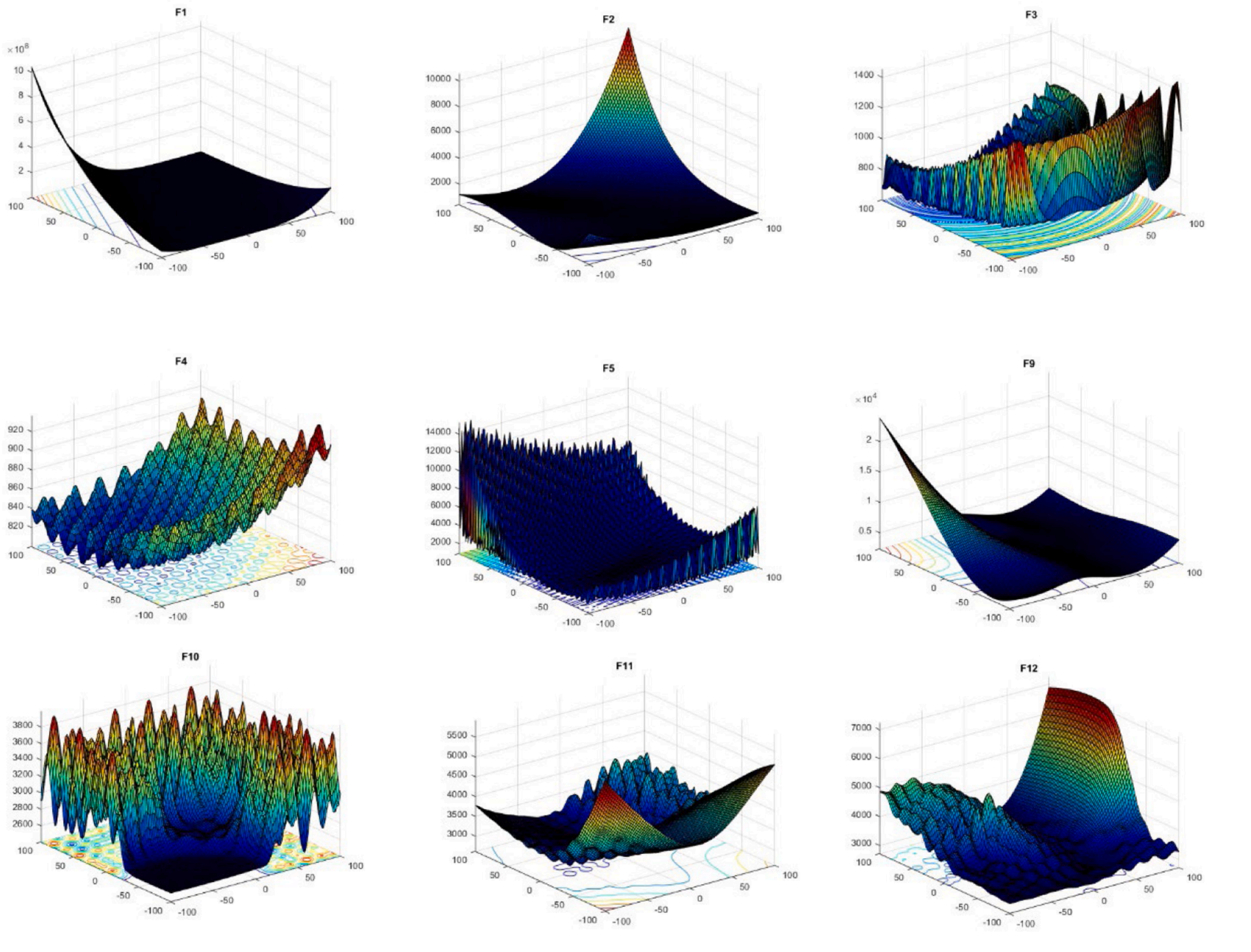


Fig. 2. 3D view of some randomly choosing CEC-2022 benchmark functions.

Table 3
Parameters of algorithms.

Algorithm	Year	Parameters values	Ref.
SOA	2019	$f_c = 2$, A ∈ [0,1]	(Dhiman & Kumar, 2019)
BWOA	2020	P_p=0.6, C_r=0.44, P_m=0.4.	(Hayyolalam & Kazem, 2020)
SCHO	2023	p=10, q=9, α=4.6, β=1.55, ϵ=0.003, n=0.5, ct=3.6, u=0.388, m=0.45	(Bai et al., 2023)
OOA	2023	Does not require additional parameter	(Dehghani & Trojovský, 2023)
WOA	2016	a ∈ [0,1]	(Mirjalili & Lewis, 2016)
GGO	2024	α=0.99, β=0.01	(El-kenawy et al., 2024)
HLOA	2024	Does not require additional parameter	(Peraza-Vázquez et al., 2024)
iHLOA	-	Does not require additional parameter	This paper

(GGO) (El-kenawy et al., 2024), and the basic HLOA (Peraza-Vázquez et al., 2024). The parameter values for all compared algorithms were obtained from their original sources. Table 3 provides the specific parameter settings used for these algorithms.

All simulations were conducted on a standardized computing platform featuring an AMD Ryzen 5 5500H CPU (3.6 GHz), Nvidia GeForce graphics card, 8 GB of 3200 MHz RAM, a 64-bit operating system, and MATLAB 2023a. This setup ensured rigorous and reliable evaluation. Consistent parameters were applied, including a population size (N) of 50, a maximum number of iterations (T) of 1000, 30 independent algorithm runs, and a problem dimension (D) of 20. Each algorithm was thoroughly evaluated, analyzing the best (minimum), worst (maximum), average, and standard deviation of their achieved results. The algorithms were ranked (Rank) based on their average performance to facilitate a meaningful comparison. In cases where average performances were identical, the standard deviation served as a tiebreaker, with a lower standard deviation indicating greater consistency. The evaluation criteria for the minimization problem are defined as follows:

- The best value

$$Best = \min(fitness_1, fitness_2, \cdots, fitness_{runs}), \tag{21}$$

- The worst value

$$Worst = \max(fitness_1, fitness_2, \cdots, fitness_{runs}), \tag{22}$$

- The average value

$$Mean = \frac{1}{runs}\sum_{i=1}^{runs} fitness_i, \tag{23}$$

- The standard deviation

$$Std = \sqrt{\frac{1}{runs-1}\sum_{i=1}^{runs}(fitness_i - Mean)^2}, \tag{24}$$

- Rank

This criterion indicates the ranking result of different algorithms. In this study, the average value mentioned above serves as the basis for ranking. If two algorithms have the same average value, the standard deviation is utilized for ranking. A smaller rank signifies better performance of the corresponding algorithm in terms of accuracy and stability.

The calculation formula is as follows:

$$Rank_i = \frac{1}{k}\sum_{k=1}^{K} rank_i^k \tag{25}$$

where K denotes the number of algorithms compared, runs is the total number of runs executed for each algorithm, while fitness represents the best fitness value of a single run. Table 4 summarizes these comparative findings.

5.3. Statistical results

This subsection provides an in-depth analysis and comparison to evaluate the performance of iHLOA against the original HLOA and several other prominent optimization algorithms, including SOA, BWOA, SCHO, OOA, WOA, and GGO. Table 4 presents the best, average, worst, and standard deviation values for iHLOA and its competitors across 12 benchmark functions with a dimensionality of 20. The best (minimum) values are highlighted in bold for emphasis. Along with the Friedman rank, these statistical metrics provide a robust framework for assessing and comparing the algorithms' performance on various optimization tasks. As presented in Table 4, the proposed iHLOA achieved the highest ranking (Rank 1) for seven benchmark functions (F2, F3, F4, F5, F7, F8, and F11). It secured second place (Rank 2) for functions F1, F9, F10, and F12. Only for function F6 iHLOA ranked third (Rank 3) among the twelve functions tested. This outstanding performance, with top two rankings in eleven out of twelve functions, demonstrates iHLOA's superior capability in finding optimal solutions. Additionally, iHLOA's overall ranking across all functions is first, with an average ranking of 1.5, followed by SOA and SCHO. These results underscore iHLOA's exceptional performance in handling complex optimization problems, indicating a robust and versatile optimization capability with superior generalization abilities.

Fig. 3 presents radar charts that illustrate the ranking performance of various algorithms across twelve test functions (F1 to F12). The radar chart for iHLOA shows the smallest enclosed area, indicating its superior performance and effectiveness in solving the optimization problems. A smaller area on the radar chart corresponds to lower ranking values, reflecting consistent and competitive performance across all test functions. In contrast, algorithms such as GGO, OOA, and BWOA exhibit larger enclosed areas, suggesting comparatively weaker performance and lower rankings across multiple functions. For instance, GGO displays a widespread area, signifying its lower efficiency and inconsistent results. Similarly, OOA and BWOA show uneven performance, struggling with certain functions. Other algorithms, such as SCHO, SOA, and WOA, demonstrate moderate performance, with their radar charts displaying a balanced spread but still enclosing larger areas compared to iHLOA. Notably, the radar chart for the original HLOA reveals significant improvements in its improved version (iHLOA), as evidenced by the reduced area and better performance rankings. This visual representation clearly highlights the outstanding competitiveness and robustness of iHLOA in comparison to the tested algorithms across all twelve functions, reaffirming its ability to outperform existing methods in diverse optimization challenges.

To further validate iHLOA's performance, a Wilcoxon rank sum test (Wilcoxon, 1992) was employed to assess the statistical significance of its results across various functions, conducted at a significance level (α) of 0.05. Table 5 summarizes the statistical outcomes when comparing the optimal solutions generated by iHLOA with seven other algorithms. This table provides insights into iHLOA's relative performance and its statistical reliability. Importantly, the results indicate significant superiority of iHLOA compared to OOA and GGO across all twelve CEC-2022 test functions (p-value < 0.05 denotes a significant difference). Similarly, compared to BWOA, iHLOA achieves statistically better results in ten out of twelve functions. Compared to standard HLOA and WOA, iHLOA demonstrates statistically significant improvements in eight and

Table 4
Comparison of iHLOA and its competitors in solving the CEC-2022 functions.

F	Index	SOA	HLOA	iHLOA	BWOA	SCHO	OOA	WOA	GGO
1	Best	3.9141E+02	**3.0000E+02**	3.0515E+02	8.8987E+02	4.0028E+02	3.1625E+03	3.0639E+03	1.5720E+03
	Worst	5.6918E+03	**3.0348E+02**	3.9611E+02	2.2446E+04	9.2724E+03	1.0613E+04	4.0248E+04	2.0985E+04
	Mean	1.0699E+03	**3.0020E+02**	3.3881E+02	8.3798E+03	3.5632E+03	7.7947E+03	1.5765E+04	8.8607E+03
	Std	1.2262E+03	**6.6333E-01**	2.0228E+01	5.3024E+03	2.6860E+03	1.7795E+03	9.5216E+03	4.9484E+03
	Rank	3	**1**	2	6	4	5	8	7
2	Best	4.0054E+02	**4.0000E+02**	4.0007E+02	4.0969E+02	4.0023E+02	4.9500E+02	4.0006E+02	4.2022E+02
	Worst	7.4357E+02	4.8847E+02	**4.8160E+02**	8.2773E+02	6.0886E+02	2.4143E+03	4.9715E+02	1.5169E+03
	Mean	4.7221E+02	4.2073E+02	**4.1693E+02**	5.5197E+02	4.4298E+02	1.0767E+03	4.3295E+02	7.4638E+02
	Std	1.0569E+02	3.0357E+01	**2.6759E+01**	1.0490E+02	5.3223E+01	4.8251E+02	3.3069E+01	2.3606E+02
	Rank	5	2	**1**	6	4	8	3	7
3	Best	6.0224E+02	6.2055E+02	**6.0171E+02**	6.1729E+02	6.0242E+02	6.2162E+02	6.1038E+02	6.1104E+02
	Worst	**6.2067E+02**	6.6747E+02	6.3193E+02	6.6697E+02	6.4569E+02	6.5489E+02	6.6530E+02	6.4584E+02
	Mean	6.1172E+02	6.4273E+02	**6.1128E+02**	6.3675E+02	6.1533E+02	6.3823E+02	6.3658E+02	6.2647E+02
	Std	**4.9026E+00**	1.3007E+01	7.0266E+00	1.2527E+01	9.4613E+00	9.1322E+00	1.2555E+01	8.9245E+00
	Rank	2	8	**1**	6	3	7	5	4
4	Best	8.0845E+02	8.1293E+02	**8.0505E+02**	8.1674E+02	8.2084E+02	8.3134E+02	8.0810E+02	8.0935E+02
	Worst	8.4206E+02	8.7164E+02	**8.3693E+02**	8.5674E+02	8.6479E+02	8.6842E+02	8.7869E+02	8.5363E+02
	Mean	8.2245E+02	8.3924E+02	**8.2062E+02**	8.4166E+02	8.3557E+02	8.4566E+02	8.4021E+02	8.3067E+02
	Std	8.1991E+00	1.5306E+01	**7.9232E+00**	9.4022E+00	1.0681E+01	8.7598E+00	1.7490E+01	1.1957E+01
	Rank	2	5	**1**	7	4	8	6	3
5	Best	9.0426E+02	9.9571E+02	**9.0032E+02**	1.0379E+03	9.3743E+02	1.0555E+03	9.9643E+02	9.0435E+02
	Worst	1.1066E+03	2.0600E+03	**9.3436E+02**	2.0491E+03	1.9607E+03	1.5908E+03	2.8630E+03	1.4712E+03
	Mean	9.6662E+02	1.4198E+03	**9.0666E+02**	1.3371E+03	1.3148E+03	1.2579E+03	1.4837E+03	1.1133E+03
	Std	3.7950E+01	2.2841E+02	**8.9669E+00**	2.6312E+02	2.5978E+02	1.2605E+02	4.4385E+02	1.7401E+02
	Rank	2	7	**1**	6	5	4	8	3
6	Best	4.9327E+03	**1.8179E+03**	1.9082E+03	1.8838E+03	2.7397E+03	2.1621E+03	1.9862E+03	1.9312E+03
	Worst	4.2982E+04	**8.1455E+03**	8.8538E+03	8.2231E+03	3.4869E+04	1.5502E+07	8.1972E+03	3.1747E+08
	Mean	1.7195E+04	**2.6087E+03**	3.7422E+03	3.8048E+03	9.3112E+03	9.4926E+05	3.7337E+03	2.0008E+07
	Std	8.4781E+03	**1.8629E+03**	1.8918E+03	2.3521E+03	6.1967E+03	3.1248E+06	1.7123E+03	6.0066E+07
	Rank	6	**1**	3	4	5	7	2	8
7	Best	2.0218E+03	2.0383E+03	**2.0214E+03**	2.0366E+03	2.0056E+03	2.0300E+03	2.0237E+03	2.0282E+03
	Worst	2.0678E+03	2.1800E+03	**2.0666E+03**	2.2137E+03	2.1157E+03	2.1395E+03	2.1326E+03	2.1041E+03
	Mean	2.0411E+03	2.1067E+03	**2.0399E+03**	2.1011E+03	2.0446E+03	2.0776E+03	2.0665E+03	2.0605E+03
	Std	1.3692E+01	3.7933E+01	**1.3081E+01**	4.6208E+01	3.0067E+01	2.2757E+01	2.6540E+01	2.0104E+01
	Rank	2	8	**1**	7	3	6	5	4
8	Best	2.2178E+03	2.2232E+03	**2.2069E+03**	2.2234E+03	2.2086E+03	2.2229E+03	2.2150E+03	2.2223E+03
	Worst	**2.2321E+03**	2.4552E+03	2.2355E+03	2.3650E+03	2.3474E+03	2.2412E+03	2.2451E+03	2.3468E+03
	Mean	2.2266E+03	2.2736E+03	**2.2261E+03**	2.2694E+03	2.2335E+03	2.2309E+03	2.2312E+03	2.2462E+03
	Std	**2.5744E+00**	6.5346E+01	5.1879E+00	5.5681E+01	3.0374E+01	4.2541E+00	5.9021E+00	3.5552E+01
	Rank	2	8	**1**	7	5	3	4	6
9	Best	2.5298E+03	**2.5293E+03**	2.5297E+03	2.5480E+03	**2.5293E+03**	2.6579E+03	2.5297E+03	2.6306E+03
	Worst	**2.6762E+03**	**2.6762E+03**	2.6764E+03	2.8957E+03	2.7429E+03	2.7906E+03	2.6764E+03	2.7699E+03
	Mean	2.5680E+03	**2.5369E+03**	2.5432E+03	2.6811E+03	2.6008E+03	2.7331E+03	2.5642E+03	2.6937E+03
	Std	4.3926E+01	**2.7201E+01**	2.7230E+01	6.6141E+01	4.5725E+01	3.5365E+01	3.6565E+01	3.4281E+01
	Rank	4	**1**	2	6	5	8	3	7
10	Best	**2.5004E+03**	2.5005E+03	2.5003E+03	2.5009E+03	2.5003E+03	2.5087E+03	2.5003E+03	2.5016E+03
	Worst	**2.5057E+03**	4.1261E+03	2.6153E+03	2.6849E+03	2.9894E+03	2.7645E+03	2.6530E+03	3.7016E+03
	Mean	**2.5009E+03**	2.7649E+03	2.5044E+03	2.5733E+03	2.6089E+03	2.6307E+03	2.5551E+03	2.6093E+03
	Std	**9.8519E-01**	4.8639E+02	2.0947E+01	7.4862E+01	1.5424E+02	7.7002E+01	6.7962E+01	2.1755E+02
	Rank	**1**	8	2	4	5	7	3	6
11	Best	2.6845E+03	**2.6000E+03**	2.6034E+03	2.6442E+03	2.6088E+03	2.8068E+03	2.6056E+03	2.7600E+03
	Worst	3.3200E+03	3.2266E+03	3.2071E+03	4.0565E+03	**3.1869E+03**	4.0489E+03	3.2108E+03	3.8428E+03
	Mean	2.8132E+03	2.7462E+03	**2.7253E+03**	3.0819E+03	2.7958E+03	3.2970E+03	2.8018E+03	3.0824E+03
	Std	1.6659E+02	2.0821E+02	1.7683E+02	3.6694E+02	1.6435E+02	3.6877E+02	**1.5443E+02**	2.9984E+02
	Rank	5	2	**1**	6	3	8	4	7
12	Best	**2.8596E+03**	2.8680E+03	2.8652E+03	2.8644E+03	2.8636E+03	2.9073E+03	2.8626E+03	2.9102E+03
	Worst	**2.8657E+03**	3.1251E+03	2.9820E+03	3.0854E+03	2.9733E+03	3.2692E+03	3.0313E+03	3.1575E+03
	Mean	**2.8637E+03**	2.9136E+03	2.8915E+03	2.9065E+03	2.8969E+03	3.0319E+03	2.8924E+03	3.0046E+03
	Std	**1.7608E+00**	5.9936E+01	3.0301E+01	4.9233E+01	2.6723E+01	9.0200E+01	4.1818E+01	5.9220E+01
	Rank	**1**	6	2	5	4	8	3	7
Mean Rank		2.9167	4.7500	**1.5000**	5.8333	4.1667	6.5833	4.5000	5.7500
Rank		2	5	**1**	7	3	8	4	6

Bold values represent the best results.

nine functions, respectively. While iHLOA outperforms SCHO and SOA in eight and six functions, respectively, the statistical significance varies across cases. These findings strongly indicate that iHLOA's superior performance compared to other algorithms is statistically significant in many instances, underscoring its robustness and effectiveness across diverse optimization scenarios.

5.4. Convergence analysis

Convergence curves provide a visual representation of the optimization process, illustrating how solutions evolve over iterations and reflecting the effectiveness of each algorithm. Fig. 4 compares the convergence trends of iHLOA with those of its competitors on the CEC-2022 functions. iHLOA exhibits diverse convergence behaviors across different functions, highlighting its adaptability to various optimization landscapes. For specific functions such as F1, F4, F5, and F10, iHLOA