RESEARCH ARTICLE

Enhanced Global Optimization Using a Novel Hybrid Sine Cosine-Gazelle Algorithm with Brownian Motion and Lévy Flight Mechanisms

Gyan Singh[1] · Saptadeep Biswas[1] · Absalom El-Shamir Ezugwu[2] · Vladimir Simic[3,4,5] · Uttam Kumar Bera[1] · Kashif Saleem[6] · Laith Abualigah[7]

Received: 14 August 2024 / Revised: 14 January 2025 / Accepted: 1 April 2025

Abstract

Metaheuristic algorithms are crucial for solving intricate optimization problems in diverse fields. The Sine Cosine Algorithm (SCA), known for its efficiency and simplicity in global search, sometimes struggles with premature convergence and inadequate exploitation. To address these challenges, this study introduces a novel hybrid SCA-Gazelle Optimisation Algorithm (HSCAGOA) by integrating the Gazelle Optimisation Algorithm's (GOA) exploitation strategy. Inspired by gazelle behaviour, GOA enhances local search capabilities, improving the balance between the exploration and exploitation phases. Additionally, HSCAGOA incorporates Brownian motion and Lévy flight mechanisms to further enhance exploration capabilities. This research rigorously evaluates HSCAGOA through extensive computational experiments on 33 benchmark test problems and six engineering design challenges. Comparisons with classical SCA and various state-of-the-art optimisation algorithms show that HSCAGOA consistently achieves faster convergence and higher solution quality across diverse optimisation landscapes. To validate these results, ranking analysis is conducted using the Wilcoxon rank-sum test and the Wilcoxon signed-rank test, confirming the efficacy of HSCAGOA. Furthermore, employing the Combined Compromise Solution (CoCoSo) method for multi-criteria decision-making enables systematic ranking and comparison of HSCAGOA's performance against other algorithms. Additionally, a comparative analysis was conducted against renowned CEC competition winners, including LSHADEcnEpSin, LSHADESPACMA, and CMA-ES. HSCAGOA is evaluated through extensive computational experiments involving CEC 2017 benchmark test functions. Moreover, sensitivity analysis is performed to assess the robustness of HSCAGOA under varying configurations, including different population sizes and maximum iteration counts on CEC 2022 benchmark test functions. The findings highlight the algorithm's adaptability and reliability in addressing complex optimization challenges. In summary, this study introduces HSCAGOA as a robust optimisation framework that mitigates the limitations of traditional SCA. It provides an effective solution for addressing complex real-world optimisation problems across different domains.

Keywords Sine Cosine Algorithm (SCA) · Gazelle Optimization Algorithm (GOA) · Global Optimization · Hybridization · Exploration and Exploitation Balance · Combined Compromise Solution (CoCoSo) method

 https://doi.org/10.1007/s44196-025-00823-6

Published online: 08 May 2025

1 Introduction

Optimization problems are widespread across fields like structural design [14], agricultural irrigation [41], economic load dispatch [50], and land map extraction accuracy [20], often involving numerous variables and complex constraints such as non-differentiable and non-convex objective functions. Traditional optimization methods face significant challenges in handling these complexities. In response, derivative-free population-based stochastic algorithms have emerged as robust solutions [39]. These algorithms offer distinct advantages over conventional methods, including simplicity, excellent global optimization capabilities, ease of implementation, and high flexibility. Consequently, they have become pivotal in optimization research over the past two decades. The process of optimization aims to find the best possible solution under specified constraints and is integral to disciplines such as science, engineering, business, and economics. Real-life optimization problems are typically formulated mathematically, and while deterministic methods exist for solving them, these are limited to problems that exhibit specific mathematical properties like continuity, differentiability, and convexity [40]. Many real-world optimization challenges do not conform to these properties, rendering traditional methods impractical. Nature-inspired algorithms, known as metaheuristic algorithms, offer a compelling alternative by treating problems as black boxes and leveraging principles inspired by natural phenomena. Their flexibility, simplicity, and ability to navigate complex solution spaces effectively make them increasingly adopted tools for tackling diverse optimization challenges.

Several well-known nature-inspired algorithms include Genetic Algorithm (GA) [19], Differential Evolution (DE) [53], Biogeography-Based Optimization (BBO) [49], Particle Swarm Optimization (PSO) [23], Artificial Bee Colony (ABC) algorithm [21], Ant Colony Optimization (ACO) [11], Grey Wolf Optimizer (GWO) [35], Ant Lion Optimizer (ALO) [31], Gravitational Search Algorithm (GSA) [45], Whale Optimization Algorithm (WOA) [34], Teaching Learning Based Optimization (TLBO) [44], Moth-flame Optimization (MFO) [30], Salp Swarm Algorithm (SSA) [33], and Harmony Search (HS) Algorithm [29]. Additionally, recently developed but widely used nature-inspired algorithms include Gradient-based optimizer (GBO) [7], Arithmetic Optimization Algorithm (AOA) [2], Archimedes optimization algorithm (AOA) [17], Snake Optimizer (SO) [18], Reptile Search Algorithm (RSA) [4], Dwarf Mongoose Optimization Algorithm (DMOA) [5], Stochastic Paint Optimization (SPO) [22], and Prairie Dog Optimization (PDO) [13]. These algorithms have achieved significant success in solving various engineering problems that traditional methods cannot address, such as feature selection, 3D mesh subdivision, and multi-modal medical image registration. A notable development in the field of nature-inspired algorithms is the No Free Lunch (NFL) theorem, which states that no single optimization algorithm is suitable for all optimization problems [55]. This theorem explains the continuous development of new algorithms and the improvement of existing ones. It emphasizes the need for continuous algorithm improvement and innovation. Metaheuristic optimization algorithms perform two essential tasks: exploration and exploitation. Exploration involves a global search to discover the most promising regions of the solution space, while exploitation focuses on refining the search in these regions to locate the optimal solution. Balancing these two tasks is crucial for the success of any metaheuristic algorithm. The NFL theorem reinforces the necessity for continuous improvement in optimization algorithms.

Recent advancements in hybrid and multi-strategy metaheuristic algorithms showcase a growing trend towards enhancing optimization capabilities across various applications. Li et al. (2022) introduced the Hybrid Multi-Strategy improved Wild Horse Optimizer (HMSWHO), which integrates diverse strategies such as the Halton sequence for population initialization and an adaptive parameter to balance exploration and exploitation, significantly improving convergence speed and accuracy [27]. Similarly, Liu et al. [28] developed a reinforcement learning-based hybrid algorithm combining the Aquila Optimizer and an improved Arithmetic Optimization Algorithm, allowing for dynamic selection between the two methods to enhance efficiency and accuracy in solving global optimization problems [28]. Uzer et al. [54] contributed to this field by proposing five hybrid algorithms that merge the Whale Optimization Algorithm with Particle Swarm Optimization and the Lévy flight algorithm, demonstrating superior performance in 19 out of 23 mathematical optimization challenges [54]. Kumari et al. [25] tackled the capacitated vehicle routing problem using a novel hybrid technique that combines the genetic algorithm with ruin and recreates strategy, achieving high-quality solutions across various benchmark instances. Lastly, Brajević et al. [10] presented a Hybrid Sine Cosine Algorithm, which integrates the sine cosine algorithm with an artificial bee colony approach to address complex engineering design optimization problems effectively. These studies illustrate a significant evolution in hybrid metaheuristic methodologies, emphasizing their effectiveness in tackling complex optimization challenges through innovative combinations of existing algorithms.

One notable metaheuristic algorithm is the Sine Cosine Algorithm (SCA) [32], developed by Seyedali Mirjalili in 2016. When handling global optimisation challenges, the SCA is well known for being effective and efficient. The SCA performs better numerically than several well-known algorithms; these benefits have led to a great deal of interest in and application of the SCA to a variety of engineering optimisation problems, including image binarization, object tracking, feature selection, data clustering, training feed-forward neural networks, optimising support vector parameters, and conceptual design of automotive components. However, like other population-based stochastic algorithms, the SCA has significant weaknesses, particularly in population diversification and balancing exploration and exploitation phases. These shortcomings often result in low exploitation capability, premature convergence, and slow convergence speed, especially in complex scenarios. This current study suggests an enhanced version of the SCA by including the exploitation mechanism of the Gazelle Optimisation Algorithm (GOA) [6] to overcome these constraints. The GOA, inspired by the hunting and evasion strategies of gazelles, enhances local search capabilities, making it an effective tool for fine-tuning solutions near the global optimum.

The primary motivation for this study is to address the inherent limitations of SCA, particularly its deficiencies in population diversification and balancing the exploration and exploitation phases. The position-update equation in the classical SCA utilises the best current individual at random to estimate the distance to the next searching zone, resulting in high exploration but weak exploitation. Despite the SCA's simplicity and efficiency, these weaknesses often result in premature convergence and slow convergence speed in complex scenarios. To overcome these insufficiencies, we propose a hybrid algorithm, the Hybrid SCA-GOA optimizer (HSCAGOA), combining the SCA with the GOA's exploitation strategy. By integrating the exploitation mechanism of GOA, which enhances local search capabilities, we aim to develop a more robust and effective optimization algorithm. This integration enhances the balance between exploration and exploitation, improving convergence speed and solution accuracy. The proposed HSCAGOA algorithm has been rigorously tested through computational experiments and comparisons on 33 benchmark test problems and engineering problems. The results demonstrate that HSCAGOA outperforms the classical SCA and other well-known metaheuristic algorithms, offering a more effective solution for global optimization challenges. This improved version, termed HSCAGOA, seeks to achieve better performance in global optimization problems by enhancing the balance between exploration and exploitation, thus improving convergence speed and solution accuracy. The ultimate goal is to provide a powerful tool for solving complex, real-world problems in various domains.

This paper is organised as follows: We begin with an introduction (Sect. 1) that provides an overview of the significance of optimisation algorithms and introduces the motivation behind developing HSCAGOA. Following this, Sect. 2 reviews existing literature on SCA and its hybridizations. The next section (Sect. 3) delves into the foundational concepts of the SCA and GOA. The core of our contribution lies in Sect. 4, where we detail the Hybrid SCA-GOA Optimizer. Moving forward, Sect. 5 presents our experimental verification and analysis. In Sect. 6, we conduct a thorough analysis and discussion of the experimental results. Finally, Sect. 7 summarises the key outcomes, discusses practical implications, and suggests future research directions.

2 Related Works

Despite its effectiveness, SCA, like many population-based metaheuristic algorithms, face challenges such as premature convergence and difficulty in maintaining population diversity. These limitations hinder its performance in complex optimization landscapes where balancing exploration to discover new regions and exploitation to refine solutions near the global optimum is crucial. To enhance the capabilities of SCA, researchers have explored various hybridization strategies, combining SCA with other metaheuristic algorithms to leverage their complementary strengths. The necessity for hybridization arises from the inherent weaknesses of standalone SCA in scenarios requiring robust population diversity maintenance and efficient exploitation of promising solution regions.

Singh and Singh [52] proposed a hybrid approach combining SCA with the Grey Wolf Optimizer (GWO), aiming to improve SCA's exploitation capabilities by integrating GWO's powerful search mechanisms. The GWO's leadership hierarchy and hunting strategy, inspired by grey wolves' social behaviour, complemented SCA's exploration strengths. This hybridization demonstrated enhanced performance on benchmark functions and real-world applications, underscoring the potential benefits of integrating complementary algorithms to enhance optimization performance. Nenavath and Jatoth [38] developed a hybrid SCA-DE algorithm to address SCA's tendency for premature convergence and to improve its robustness in navigating complex optimization landscapes. Differential Evolution, known for its population diversity maintenance and global exploration capabilities, was integrated with SCA to enhance solution quality and convergence speed. Evaluations across diverse benchmark functions highlighted significant improvements over traditional SCA and standalone DE, reinforcing the effectiveness of hybrid approaches in tackling complex optimization challenges.

Despite these advancements, challenges such as maintaining a balance between exploration and exploitation persist in hybrid SCA variants. The Gazelle Optimization Algorithm (GOA), inspired by the evasive strategies of gazelles in the wild, offer a promising solution to enhance SCA's exploitation capabilities [6]. GOA's adaptive search mechanism and efficient local exploration strategies make it particularly suitable for fine-tuning solutions near the global optimum, addressing one of SCA's primary weaknesses. In response to the limitations of existing hybrid SCA variants, this paper proposes the Hybrid SCA-GOA optimizer (HSCAGOA). The integration of GOA's exploitation mechanism aims to augment SCA's exploration-exploitation balance, improving convergence speed and solution accuracy across diverse optimization scenarios. HSCAGOA combines SCA's simplicity and efficiency with GOA's adaptive local search strategies, offering a robust solution for complex optimization landscapes where traditional methods falter. The motivation for developing HSCAGOA lies in its potential to overcome the inherent limitations of both standalone SCA and existing hybridizations. By integrating GOA's exploitation capabilities, HSCAGOA seeks to achieve superior performance in terms of convergence speed and solution quality, thereby addressing critical challenges in global optimization. Computational experiments and comparisons on benchmark functions and engineering problems will validate the effectiveness of HSCAGOA, providing empirical evidence of its superiority over traditional SCA and other metaheuristic algorithms.

Recent advancements in SCA research have led to several hybrid algorithms that enhance its performance across various domains. Gupta et al. [15] developed a modified SCA (MSCA) featuring a non-linear transition rule and a mutation operator, significantly improving performance on benchmark problems and real-engineering scenarios. Singh and Kaur [51] combined SCA with Harmony Search (HS) to form HSCAHS, overcoming SCA's convergence shortcomings and achieving efficient optimization in various design problems. Abualigah and Dulaimi [3] introduced SCAGA, a hybrid of SCA and Genetic Algorithm for feature selection, showing superior performance on UCI machine learning datasets. Altay and Alatas [8] proposed hybrid methods combining Differential Evolution and SCA for association rule mining, demonstrating superior performance on specific datasets. Seyyedabbasi [48] introduced WOASCALF, combining the Whale Optimization Algorithm (WOA), SCA, and Levy Flight. This hybridization demonstrated strong exploration abilities, outperforming other algorithms on benchmark functions and engineering problems [48]. Abdel-Mawgoud et al. [1] developed a hybrid approach combining the Arithmetic Optimization Algorithm (AOA) with SCA for Battery Energy Storage (BES) integration in distribution networks. Their method achieved significant reductions in power losses, highlighting the effectiveness of integrating SCA with arithmetic optimization techniques [1].

Duan and Yu [12] proposed cHGWOSCA, integrating GWO with SCA. This collaboration enhanced GWO's position update process with SCA's exploration abilities, demonstrating superior performance in engineering optimization problems [12]. Pham et al. [42] introduced nSCA, integrating roulette wheel selection (RWS) and opposition-based learning (OBL) with SCA. Their hybrid approach showed superiority in both theoretical analyses and practical engineering optimization problems, emphasizing the benefits of incorporating adaptive selection and learning mechanisms [42]. Li et al. [26] proposed EBSCA, enhancing SCA's exploitation capabilities through a new position-update equation and orthogonal crossover strategy. Their approach exhibited improved performance on benchmark functions and real-world engineering applications, demonstrating effective exploitation of promising solution regions [26]. Pham et al. [43] further refined nSCA with RWS and OBL, consistently delivering superior solutions in various engineering optimization challenges. Their iterative improvements underscore the robustness and adaptability of hybrid SCA variants in handling complex optimization landscapes [43]. Nadimi-Shahraki et al. [37] introduced MTV-SCA, employing multi-trial vector strategies to balance exploration and exploitation effectively. This approach mitigates premature convergence issues and demonstrates effectiveness in real-world applications, contributing to advancing the state-of-the-art in global optimization [37].

These hybridizations illustrate the evolving landscape of SCA research, focusing on addressing its inherent limitations through integration with diverse optimization techniques. The proposed HSCAGOA builds upon these advancements by leveraging the Gazelle Optimization Algorithm (GOA)'s unique capabilities, aiming to set new benchmarks in optimization performance. The development of the SCA has marked a significant advancement in the field of global optimization, offering a simple yet effective approach to navigating complex solution spaces. Hybridization strategies, as evidenced by recent research, have played a pivotal role in enhancing SCA's applicability and performance across diverse optimization challenges. The proposed Hybrid SCA-GOA optimizer (HSCAGOA) represents a promising evolution in this trajectory, integrating GOA's exploitation mechanisms to augment SCA's exploration-exploitation balance.

3 Preliminaries

Since the proposed HSCAGOA is based on the Sine Cosine Algorithm (SCA) and the Gazelle Optimization Algorithm (GOA), a brief description of the optimization mechanisms of these two algorithms is provided below.

3.1 Sine Cosine Algorithm

The Sine Cosine Algorithm (SCA), developed by Mirjalili in 2016, is known for its ability to generate a set of stochastic solutions during the search process, offering a high degree of exploration. The unique feature of SCA lies in its use of sine and cosine wave functions to control the exploration and exploitation of the solution space. The position update equations of SCA are given by:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \phi_1 \times \sin(\phi_2) \times \left|\phi_3 \times \vec{l}_i(t) - \vec{X}_i(t)\right| \tag{1}$$

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \phi_1 \times \cos(\phi_2) \times \left|\phi_3 \times \vec{l}_i(t) - \vec{X}_i(t)\right| \tag{2}$$

where $\vec{X}_i(t)$ is the position of the solution at iteration t, ϕ_1 is a linearly decreasing variable from 2 to 0, ϕ_2 is a random number uniformly distributed in $[0, 2\pi]$, ϕ_3 is a random number in the interval $(0, 2)$, and $\vec{l}_i(t)$ represents the position of the optimal solution. The variations in sine and cosine functions within the range $[-2, 2]$ allow for robust exploration beyond the destination point, enhancing SCA's exploratory capabilities.

A random number ϕ_4 uniformly distributed in $[0, 1]$ is introduced to switch between exploitation and exploration. The combined position update equation incorporating ϕ_4 is given by:

$$\vec{X}_i(t+1) = \begin{cases} \vec{X}_i(t) + \phi_1 \times \sin(\phi_2) \times \left|\phi_3 \times \vec{l}_i(t) - \vec{X}_i(t)\right|, & \text{if } \phi_4 < 0.5 \\ \vec{X}_i(t) + \phi_1 \times \cos(\phi_2) \times \left|\phi_3 \times \vec{l}_i(t) - \vec{X}_i(t)\right|, & \text{if } \phi_4 \geq 0.5 \end{cases} \tag{3}$$

3.2 Gazelle Optimization Algorithm

Gazelles inhabit drylands across various regions, including Asia, the Arabian Peninsula, the Sahara desert, the sub-Saharan Sahel, and northeast Africa. As common prey for many predators, gazelles have developed strong senses of hearing, sight, and smell to compensate for their vulnerabilities by effectively outrunning predators. Their survival strategies in nature involve several key behaviours. When predators are absent, gazelles engage in grazing, allowing them to exploit available resources. Additionally, they exhibit a unique behaviour known as stotting, where they leap up to two meters in the air to spot potential predators and alert other gazelles. In the presence of predators, they rely on their exceptional speed, with a maximum speed of up to 100 km/hr, to escape danger. These adaptive characteristics of the gazelle are modelled in the Gazelle Optimization Algorithm (GOA) to solve complex optimization problems, drawing inspiration from their natural survival strategies.

3.2.1 Population Initialization

The GOA is a population-based optimization algorithm that employs randomly initialized search agents, referred to as gazelles. These search agents are represented in an $n \times d$ matrix of candidate solutions:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d-1} & x_{n,d} \end{pmatrix}$$

Each position vector $x_{i,j}$ is generated using:

$$x_{i,j} = \text{rand} \times (\text{UB}_j - \text{LB}_j) + \text{LB}_j \tag{4}$$

where rand is a random number, and UB_j and LB_j are the upper and lower bounds of the problem, respectively. The best solutions are considered the elite gazelles, used to form an Elite $n \times d$ matrix:

$$\text{Elite} = \begin{pmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,d-1} & x'_{1,d} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,d-1} & x'_{2,d} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x'_{n,1} & x'_{n,2} & \cdots & x'_{n,d-1} & x'_{n,d} \end{pmatrix}$$

This elite matrix guides the search process and is updated at each iteration if better solutions are found.

3.2.2 Brownian Motion

Brownian motion represents random movement characterized by a Gaussian probability distribution with a mean of zero and unit variance:

$$f_{\text{B}}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{5}$$

where $\mu = 0$ and $\sigma^2 = 1$.

3.2.3 Lévy Flight

Lévy flight involves a random walk using a Lévy distribution:

$$L(x_j) \propto x_j^{-\alpha} \tag{6}$$

where x_j denotes the flight distance and $\alpha \in (1, 2)$ represents the power-law exponent. The Lévy stable process is defined as:

$$f_L(x; \alpha, \gamma) = \frac{1}{\pi} \int_0^{\infty} \exp(-\gamma q^{\alpha}) \cos(qx)\, \mathrm{d}q \tag{7}$$

where α is the distribution index and γ is the scale unit. Lévy motion is generated using:

$$\text{Levy}(\alpha) = 0.05 \times \frac{x}{|y|^{1/\alpha}} \tag{8}$$

where x and y are defined as:

$$x = \text{Normal}(0, \sigma_x^2) \quad \text{and} \quad y = \text{Normal}(0, \sigma_y^2) \tag{9}$$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2)\alpha 2^{(\alpha-1)/2}} \right]^{1/\alpha}, \quad \sigma_y = 1, \quad \alpha = 1.5 \tag{10}$$

3.2.4 Exploitation and Exploration Phases of GOA

GOA mimics the survival behaviour of gazelles through two main phases: exploitation and exploration.

Exploitation Phase

In this phase, gazelles graze without immediate predator threats, using Brownian motion for local search. The updated equation is:

$$X_i^{t+1} = X_i^t + s \times \text{RB} \times (\text{Elite}_i^t - \text{RB} \times X_i^t) \tag{11}$$

where X_i^{t+1} is the next iteration solution, X_i^t is the current solution, s denotes grazing speed, RB is a random vector, and Elite_i^t is the elite solution's position.

Exploration Phase

When a predator is detected, gazelles perform Lévy flights, modelled as:

$$X_i^{t+1} = X_i^t + S \times l \times \text{RL} \times (\text{Elite}_i^t - \text{RL} \times X_i^t) \tag{12}$$

where S is the maximum speed, and RL is a vector of random numbers based on Lévy distributions. The predator's behaviour combines Brownian motion and Lévy flights, modelled as:

$$X_i^{t+1} = X_i^t + S \times l \times \text{CF} \times \text{RB} \times (\text{Elite}_i^t - \text{RL} \times X_i^t) \tag{13}$$

where $\text{CF} = 1 - \frac{i}{\text{Max\_iter}}$. The Predator Success Rate (PSR) ensures the algorithm avoids local minima:

$$X_i^{t+1} = \begin{cases} X_i^t + \text{CF} \times (\text{LB} + R \times (\text{UB} - \text{LB})), & \text{if } r < \text{PSRs} \\ X_i^t + [\text{PSRs} \times (1-r) + r] \times X_{r1} - X_{r2}, & \text{else} \end{cases} \tag{14}$$

where LB and UB are the bounds, r is a random number, and $r1$ and $r2$ are random indices.

4 Hybrid SCA-GOA Optimizer

In the Hybrid Sine Cosine Algorithm with Gazelle Optimization Algorithm (HSCAGOA), the classical Sine Cosine Algorithm (SCA) position-update equation is modified to incorporate the exploration and exploitation behaviours inspired by gazelles. This hybrid optimizer enhances the exploration and exploitation capabilities, making it more effective in global optimization tasks.

4.1 Modified Position-Update Equation

The core of the HSCAGOA lies in its modified position-update equation, which adapts the classical SCA with additional mechanisms for exploration and exploitation. The position-update process depends on a random coefficient ϕ_4, determining whether the algorithm focuses on exploration and exploitation. The modified position-update equation is defined as follows:

When $\phi_4 < 0.5$

$$X(i, j) = \mathrm{Elite}(i, j) + S \cdot \mu \cdot \mathrm{CF} \cdot (\mathrm{RB}(i, j) \cdot (\mathrm{RL}(i, j) \cdot \mathrm{Elite}(i, j) - X(i, j))) + \phi_1 \cdot \sin(\phi_2) \cdot |\phi_3 \cdot l_i(t) - X_i(t)| \quad (15)$$

When $\phi_4 \geq 0.5$

$$X_i(t+1) = X(i, j) + s \cdot R \cdot (\mathrm{RB}(i, j) \cdot (\mathrm{Elite}(i, j) - \mathrm{RB}(i, j) \cdot X(i, j))) + \phi_1 \cdot \cos(\phi_2) \cdot |\phi_3 \cdot l_i(t) - X_i(t)| \quad (16)$$

where,

- $X(i, j)$: The current position of the ith individual in the jth dimension.
- $\mathrm{Elite}(i, j)$: The position of the elite individual in the jth dimension.
- S: A scaling factor.
- μ: A control parameter for the exploration and exploitation balance.
- CF: Control factor.
- $\mathrm{RB}(i, j)$: Random binary vector element for the ith individual in the jth dimension.
- $\mathrm{RL}(i, j)$: Random learning coefficient for the ith individual in the jth dimension.
- $\phi_1, \phi_2, \phi_3, \phi_4$: Random coefficients that influence the search direction and distance.
- $l_i(t)$: The i-th individual's position at time t.
- R: A random number in the range [0,1].

4.2 Phases

4.2.1 Exploration Phase

This phase is characterized by a higher degree of randomness and larger movements, helping the algorithm to search through the global search space. When $\phi_4 < 0.5$, the Eq. (15) includes the sine function, which induces a diverse range of solutions.

Algorithm 1: Pseudocode of Hybrid SCA-GOA algorithm

Input: Benchmark_Function_ID, N (number of search agents), Max_Iteration
Output: Destination_fitness, Destination_position, Convergence_curve

Initialization:
$X \leftarrow \text{initialization\_SCA}(N, \text{dim}, \text{up}, \text{down})$
$\text{Top\_gazelle\_pos} \leftarrow \text{zeros}(1, \text{dim})$
$\text{Top\_gazelle\_fit} \leftarrow \infty$
$\text{Convergence\_curve} \leftarrow \text{zeros}(1, \text{Max\_Iteration})$

Main Loop:
for $t = 1$ to Max_Iteration **do**
$\phi_1 \leftarrow 2 \times (1 - \frac{t}{\text{Max\_Iteration}})$
$\phi_2 \leftarrow 2 \times \pi \times \text{rand}()$, $\phi_3 \leftarrow 2 \times \text{rand}()$
$\phi_4 \leftarrow \text{rand}()$, $s \leftarrow \text{rand}()$, $R \leftarrow \text{rand}()$
$S \leftarrow 0.88$, $CF \leftarrow (1 - \frac{t}{\text{Max\_Iteration}})^{2 \times \frac{t}{\text{Max\_Iteration}}}$
$RL \leftarrow 0.05 \times \text{levy}(N, \text{dim}, 1.5)$, $RB \leftarrow \text{randn}(N, \text{dim})$
for $i = 1$ to N **do**
for $j = 1$ to dim **do**
if rand() $\phi_4 < 0.5$ **then**
$X[i,j] \leftarrow \text{Top\_gazelle\_pos}[j] + S \times \mu \times CF \times RB[i,j] \times (RL[i,j] \times \text{Top\_gazelle\_pos}[j] - X[i,j]) + (\phi_1 \times \sin(\phi_2) \times |\phi_3 \times \text{Top\_gazelle\_pos}[j] - X[i,j]|)$
else
$X[i,j] \leftarrow X[i,j] + s \times R \times RB[i,j] \times (\text{Top\_gazelle\_pos}[j] - RB[i,j] \times X[i,j]) + (\phi_1 \times \cos(\phi_2) \times |\phi_3 \times \text{Top\_gazelle\_pos}[j] - X[i,j]|)$
Evaluate fitness
if $\text{fitness}(X[i]) < \text{Top\_gazelle\_fit}$ **then**
$\text{Top\_gazelle\_pos} \leftarrow X[i]$, $\text{Top\_gazelle\_fit} \leftarrow \text{fitness}(X[i])$
$\text{Convergence\_curve}[t] \leftarrow \text{Top\_gazelle\_fit}$
Output: Destination_fitness, Destination_position (Top_gazelle_pos), Convergence_curve

4.2.2 Exploitation Phase

This phase focuses on refining the solutions by narrowing down the search area around the best-found solutions. When $\phi_4 \geq 0.5$, the Eq. (16) includes the cosine function, leading to more fine-tuned adjustments around the elite solutions.

This hybrid approach leverages the global search capabilities of the Sine Cosine Algorithm and the local search and optimization traits inspired by the gazelle's adaptive behaviours, enhancing the overall optimization performance.

4.3 Incorporation of Brownian Motion and Lévy Flight

Brownian motion is a random walk where the step size follows a normal distribution. It is used in the HSCAGOA to provide small, random perturbations to the positions of individuals, which helps in fine-tuning solutions and avoiding local optima.

The random binary vector RB follows a normal distribution:

$$\text{RB}(i, j) \sim \mathcal{N}(0, 1)$$

Lévy flight is a random walk where the step sizes follow a Lévy distribution, characterized by occasional large steps. It enhances the exploration capability by allowing the algorithm to take large jumps, which can help in discovering new regions of the solution space.

 https://doi.org/10.1007/s44196-025-00823-6

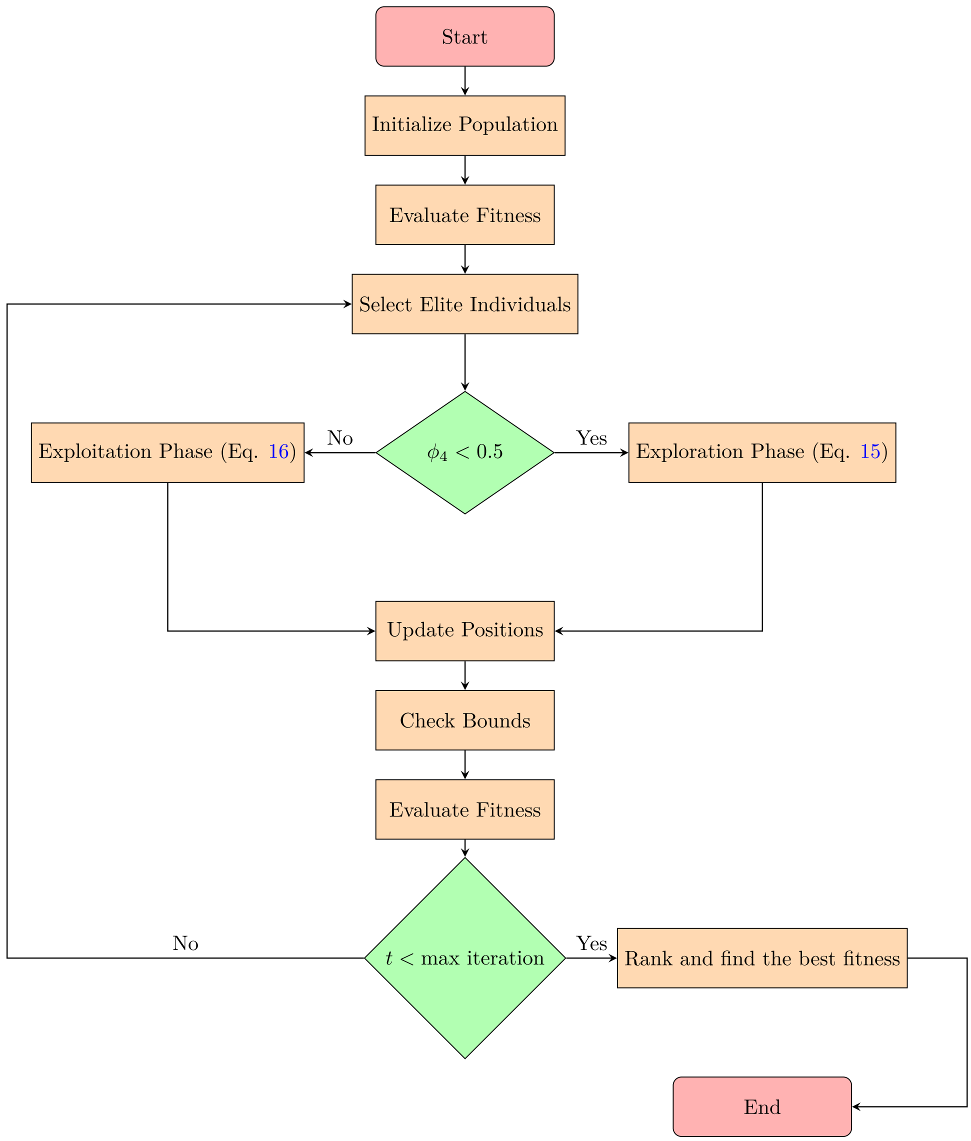


Fig. 1 Flowchart of HSCAGOA algorithm

The random learning coefficient RL follows a Lévy distribution:

$$\mathrm{RL}(i, j) \sim \text{Lévy}(\lambda)$$

In HSCAGOA, these mechanisms are integrated as follows:

$$\begin{aligned}\mathrm{RL} &= 0.05 \times \text{Lévy}(1.5)\\ \mathrm{RB} &= \mathrm{randn}(sca\_no, dim)\end{aligned}$$

The process of HSCAGOA is outlined in the pseudocode shown in Algorithm 1. This pseudocode demonstrates the initialization, main loop, exploration, exploitation phases, and fitness evaluation procedures that drive the algorithm. To provide a visual overview of the algorithm's workflow, the flowchart in Fig. 1 illustrates the key steps, from initialization to evaluating the fitness and ranking of the best solutions.

Table 1 Uni-modal benchmark functions overview

Function	Description	Dimensions	Range	$f_{\min}$
Sphere (F1)	$g(z) = \sum_{i=1}^{n} z_i{}^2$	30	[−100,100]	0
Schwefel 2.22 (F2)	$g(z) = \sum_{i=0}^{n} \lvert z_i\rvert + \prod_{i=0}^{n} \lvert z_i\rvert$	30	[−10,10]	0
Schwefel 1.2 (F3)	$g(z) = \sum_{i=1}^{d} (\sum_{j=1}^{i} z_i)^2$	30	[−100,100]	0
Schwefel 2.21 (F4)	$g(z) = \max_i\{\lvert z_i\rvert, 1 \leq i \leq n\}$	30	[−100,100]	0
Rosenbrock (F5)	$g(z) = \sum_{i=1}^{n-1} [100(z_i{}^2 - z_{i+1})^2 + (1 - z_i)^2]$	30	[−30,30]	0
Step (F6)	$g(z) = \sum_{i=1}^{n} ([z_i + 0.5])^2$	30	[−100,100]	0
Quartic (F7)	$g(z) = \sum_{i=1}^{n} i z_i{}^4 + random[0, 1)$	30	[−1.28,1.28]	0

4.4 The Computational Complexity of the HSCAGOA

The computational complexity of the HSCAGOA optimizer is analyzed in terms of space and time complexities. Starting with the space complexity, the primary factors contributing to the storage requirements are the population of solutions and associated variables. Let n represent the number of solutions in the population, d denote the dimensionality of each solution, and T signify the number of iterations. The main components influencing the space complexity include the population matrix X with a space requirement of $O(n \times d)$, the fitness values for each solution requiring $O(n)$, elite solutions necessitating $O(n \times d)$, the step size matrix also contributing $O(n \times d)$, and other temporary variables like intermediate fitness values and normalization factors that add $O(n \times d)$ to the overall complexity. Therefore, the space complexity of the HSCAGOA optimizer is $O(n \times d)$.

The time complexity of the HSCAGOA optimizer is driven by the number of operations conducted during each iteration and the total number of iterations. In each iteration, several key steps are performed. First, the fitness evaluation of all n solutions involves a complexity of $O(n \times f)$, where f is the complexity of the fitness function. Next, selecting the elite solutions involves sorting, which requires $O(n \log n)$. The position updates of all n solutions across d dimensions entail $O(n \times d)$, and ensuring that solutions remain within bounds adds another $O(n \times d)$ to the complexity. These operations are repeated across T iterations, resulting in an overall time complexity of $O(T \times (n \times f + n \log n + n \times d + n \times d))$. The dominant terms are $n \times f$ and $n \times d$, leading to a time complexity of $O(T \times (n \times f + n \times d))$. Since f is typically more significant than d in most optimization problems, the time complexity can be approximated as $O(T \times n \times f)$, where T is the number of iterations, n is the population size, and f is the complexity of the fitness function.

5 Experimental Verification and Analysis

This section outlines the experimental design used to evaluate the performance of the HSCAGOA. The evaluation utilized twenty-eight CEC 2017 test functions, five classical test functions, and six selected optimization design problems from the engineering domain. The performance of the HSCAGOA was compared with several algorithms, including the Sine Cosine Algorithm (SCA) [32], Differential Evolution (DE) [53], Gravitational Search Algorithm (GSA) [45], Constriction Coefficient-Based Particle Swarm Optimization and Gravitational Search Algorithm (CPSOGSA) [46], Biogeography-Based Optimization (BBO) [49], Arithmetic Optimization Algorithm (AOA) [2], Dwarf Mongoose Optimization Algorithm (DMOA) [5], Prairie Dog Optimization Algorithm (PDO) [13], Grey Wolf Optimizer (GWO) [35], and Salp Swarm Algorithm (SSA) [33]. The HSCAGOA algorithm was implemented using MATLAB R2022b to evaluate its performance on selected benchmarks and engineering problems. The experiments were conducted on a workstation running Windows 11 Pro, equipped with an Intel (R) Xeon (R) W7-2495X @ 2.50 GHz processor and 128 GB of RAM. Key parameters, such as the population size and the maximum number of iterations, were carefully fine-tuned for this study. The population size was set to 30, and the maximum number of iterations was set to 1000.

 https://doi.org/10.1007/s44196-025-00823-6

Table 2 Multi-modal benchmark functions overview

Function	Description	Dimensions	Range	$f_{\min}$
Schwefel (F8)	$g(z) = \sum_{i=1}^{n}(-z_i \sin(\sqrt{\lvert z_i \rvert}))$	30	[−500,500]	$-418.9829 \times n$
Rastrigin (F9)	$g(z) = \sum_{i=1}^{n}[{z_i}^2 - 10\cos(2\pi z_i) + 10]$	30	[−5.12,5.12]	0
Ackley (F10)	$g(z) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}{z_i}^2})$	30	[−32,32]	0
	$-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)) + 20 + e$			
Griewank (F11)	$g(z) = 1 + \frac{1}{4000}\sum_{i=1}^{n}{z_i}^2 - \prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}})$	30	[−600,600]	0
Penalized (F12)	$g(z) = \frac{\pi}{n}\{10\sin(\pi y_i)\} + \sum_{i=1}^{n-1}(y_i - 1)^2$	30	[−50,50]	0
	$[1 + 10\sin^2(\pi y_{i+1}) + \sum_{i=1}^{n} u(z_i, 10, 100, 4)]$			
	$where\ y_i = 1 + \frac{z_i+1}{4}$,			
	$u(z_i, a, k, m) = \begin{cases} \{K(z_i - a)^m, & if z_i > a \\ 0, & if -a \le z_i \ge a \\ K(-z_i - a)^m & if -a \le z_i \end{cases}$			
Penalized 2 (F13)	$g(z) = 0.1(\sin^2(3\pi z_1) + \sum_{i=1}^{n}(z_i - 1)^2$	30	[−50,50]	0
	$[1 + \sin^2(3\pi z_i + 1)](z_n - 1)^2$			
	$\{1 + \sin^2(2\pi z_n)\}) + \sum_{i=1}^{n} u(z_i, 5, 100, 4)$			

Table 3 Benchmark functions for fixed-dimension multi-modal testing

Function	Description	Dimensions	Range	$f_{\min}$
Foxholes (F14)	$g(z) = \{(\frac{1}{500}\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}2(z_i-a_{ij})^6})\}^{-1}$	2	[−65,65]	1
Kowalik (F15)	$g(z) = \sum_{i=1}^{11}\left[a_i - \frac{z_1(b_i^2+b_i z_2)}{b_i^2+b_i z_3+z_4}\right]^2$	4	[−5,5]	0.0003
Six Hump Camel (F16)	$g(z) = 4z_1^2 - 2.1z_1^4 + \frac{1}{3}z_1^6 + z_1z_2 - 4z_2^2 + 4z_2^4$	2	[−5,5]	−1.0316
Branin (F17)	$g(z) = (z_2 - \frac{5.1}{4\pi^2}z_1^2 + \frac{5}{\pi}z_1 - 6)^2$ $+10(1-\frac{1}{8\pi})\cos z_1 + 10$	2	[−5,5]	0.398
Goldstein-Price (F18)	$g(z) = [1 + (z_1 + z_2 + 1)^2(19 - 14z_1 + 3z_1^2$ $-14z_2 + 6z_1z_2 + 3z_2^2)] \times [30 + (2z_1 - 3z_2 + 1)^2$ $(18 - 32z_1 + 12z_1^2 + 48z_2 - 36z_1z_2 + 27z_2^2)]$	2	[−2,2]	3
Hartman 3 (F19)	$g(z) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{i=1}^{3} a_{ij}(z_j - p_{ij})^2\right)$	3	[−1,2]	−3.86
Hartman 6 (F20)	$g(z) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{i=1}^{6} a_{ij}(z_j - p_{ij})^2\right)$	6	[0,1]	−0.32
Shekel 5 (F21)	$g(z) = -\sum_{i=1}^{5}[(z-a_i)(z-a_i)^T + c_i]^{-1}$	4	[0,1]	−10.1532
Shekel 7 (F22)	$g(z) = -\sum_{i=1}^{7}[(z-a_i)(z-a_i)^T + c_i]^{-1}$	4	[0,1]	−10.4028
Shekel 10 (F23)	$g(z) = -\sum_{i=1}^{10}[(z-a_i)(z-a_i)^T + c_i]^{-1}$	4	[0,1]	−10.5363
Schaffers (F24)	$g(z) = 0.5 + \frac{\sin^2(z_1^2+z_2^2)-0.5}{[1+0.001(z_1^2+z_2^2)]^2}$	2	[−100,100]	0
Easom (F25)	$g(z) = -\cos(z_1)\cos(z_2)exp[-(z_1-\pi)^2 \times 9(z_2-\pi)^2]$	2	[−100,100]	0
Schwefel 2.26 (F26)	$g(z) = -\sum_{i=1}^{n} z_i \sin(\sqrt{\lvert z_i \rvert})$	30	[−500,500]	−418.982
Bohachevsky (F27)	$g(z) = z_1^2 + 2z_2^2 - 0.3\cos(3\pi z_1) - 0.4\cos(4\pi z_2) + 0.7$	2	[−100,100]	0
Bohachevsky 3 (F28)	$g(z) = z_1^2 + 2z_2^2 - 0.3\cos(3\pi z_1) - 0.3$	2	[−50,50]	0
Colville (F29)	$g(z) = 100(z_1 - z_2^2)^2 + (1 - z_1)^2 + 90(z_4 - z_3^2)^2 + (1 - z_2)^2$ $+10.1(z_2 - 1)^2 + (z_4 - 1)^2 + 19.8(z_2 - 1)(z_4 - 1)$	4	[−10,10]	0

 https://doi.org/10.1007/s44196-025-00823-6

Table 4 Overview of fixed-dimension uni-modal benchmark functions

Function	Description	Dimensions	Range	$f_{\min}$
Booth (F30)	$g(z) = (2z_1 + z_2 - 5)^2 + (z_1 + 2z_2 - 7)^2$	2	[−10,10]	0
Matyas (F31)	$g(z) = 0.26(z_1^2 + z_2^2) - 0.48z_1z_2$	2	[−10,10]	0
Zettl (F32)	$g(z) = (z_1^2 + z_2^2 - 2z_1)^2 + 0.25z_1$	2	[−1,5]	−0.00379
Leon (F33)	$g(z) = 100(z_2 - z_1^3)^2 + (1 - z_1)^2$	2	[−1.2,1.2]	0

5.1 Benchmark Test Functions

In this section, we present a comprehensive set of benchmark test functions used to evaluate the performance of the proposed HSCAGOA. These test functions are categorized into four groups: Uni-modal Benchmark Functions, Multi-modal Benchmark Functions, Fixed-Dimension Multi-modal Test Functions, and Fixed-Dimension Uni-modal Test Functions. Each category contains a variety of functions with different characteristics, allowing for a thorough assessment of the algorithm's capabilities.

The Uni-modal Benchmark Functions focus on single-peak landscapes, which are simpler but essential for testing the algorithm's ability to converge to a global optimum. The Multi-modal Benchmark Functions include multiple peaks and valleys, making them suitable for evaluating the algorithm's exploration and exploitation balance. Fixed-dimension multi-modal and Uni-modal Test Functions are included to test the algorithm's performance on problems with a set number of variables, providing insights into its scalability and robustness.

Tables 1, 2, 3, and 4 detail the functions used, including their mathematical formulations, dimensions, ranges, and known minimum values ($f_{\min}$). These benchmark functions serve as a standardized way to compare the performance of different optimization algorithms.

5.2 Experimental Setup

This section details the experimental setup used to evaluate the performance of the HSCAGOA and compares it with other well-known optimization algorithms. The primary focus is on the algorithm control parameters, which play a critical role in determining the effectiveness and efficiency of the optimization process. Table 5 provides a comprehensive overview of the control parameters for each algorithm tested. For consistency and fair comparison, a uniform population size of 30 and a maximum iteration count of 1000 were chosen across all algorithms. Additionally, each algorithm's specific parameters were set according to the recommendations from their respective literature sources.

The control parameters of the HSCAGOA are designed to optimize the algorithm's search capabilities and enhance its performance. The parameter ϕ_1 plays a crucial role in the exploration phase, as it decreases linearly from an initial value of 2 to 0 throughout the iterations. This gradual reduction allows the algorithm to balance exploration and exploitation effectively. The parameters ϕ_2, ϕ_3, and ϕ_4 are generated as random numbers within defined ranges, introducing stochasticity into the search process and enabling the algorithm to escape local optima. Specifically, ϕ_2 is constrained between 0 and 2π, while ϕ_3 is bounded within the interval (0, 2), and ϕ_4 is limited to values between 0 and 1. Additionally, the constant parameter a is set to 2, which influences the algorithm's convergence behaviour. The selection parameter S is fixed at 0.88, impacting the decision-making process during the search. Collectively, these control parameters are pivotal in guiding HSCAGOA's optimization trajectory, enabling it to effectively navigate the solution space and achieve superior results in comparison to other algorithms.

The control parameters for the various optimization algorithms employed in this study are crucial for ensuring effective performance and comparability, as summarized in Table 5. The SCA is configured with a population size of 30 and a maximum iteration limit of 1000, using a constant parameter a set to 2. In the DE algorithm, the population size remains at 30, with a maximum iteration of 1000, along with a scaling factor ranging from 0.2 to 0.8 and a crossover probability of 0.8. The GOA also utilizes a population size of 30 and 1000 iterations, with the parameter α set to 20. For the CPSOGSA algorithm, a population size of 30 and maximum iterations of 1000 are maintained, with control parameters $\langle P_1 \rangle$ and $\langle f_2 \rangle$ fixed at 2.05. The BBO algorithm operates with a population size of 30 and a maximum iteration of 1000, featuring a mutation probability of 0.9 and a maximum inertia weight of 0.2. The AOA is set with a population size of 30 and 1000 iterations, with α fixed at 5. The DMOA uses a population size of 30 and a maximum iteration of 1000, with sensitive parameters α and β set to 0.1 and 0.005, respectively. The PDO method is configured similarly, maintaining a population size of 30 and a maximum iteration of 1000, while the GWO employs a linear decreasing strategy for a from 2 to 0. Lastly, the Salp Swarm Algorithm (SSA) utilizes random numbers for its parameters C_2 and C_3 within the range of [0,1]. These parameter settings enable each algorithm to perform optimally, ensuring a fair comparison with the proposed HSCAGOA.

5.3 Results

This section presents the performance evaluation of various optimization algorithms on both uni-modal and multi-modal test functions. The results are summarized across dimensions and algorithms, highlighting key metrics such as best, worst, average, median values, and standard deviation (SD).

Table 6 details the performance of uni-modal functions each with a dimension of 10 across different algorithms. Across these functions, algorithms SCA, DE, and CPSOGSA consistently demonstrate robust performance in minimizing objective functions, achieving global best values close to zero across multiple functions.

For instance, in F1 and F2, these algorithms consistently converge to optimal solutions with negligible deviations, as indicated by low average, median, and SD values. This stability suggests their effectiveness in handling simpler uni-modal landscapes where finding a single optimal solution is crucial. However, as the complexity of the objective functions increases, as observed in functions F3 to F7, the performance variability among algorithms becomes more apparent. Algorithms HSCAGOA excel in achieving extremely low best values in highly complex landscapes in F3, whereas GSA shows competitive results on function F5. The variability in performance metrics such as average, median, and SD across these functions underscores the diverse challenges posed by different optimization landscapes. Furthermore, algorithms such as BBO and SSA also exhibit competitive performances in certain functions, highlighting their adaptability to specific optimization requirements. The evaluated algorithms generally demonstrate stable convergence towards optimal solutions across uni-modal test functions.

The performance of various optimisation algorithms across multimodal test functions (F8–F13) is evaluated and summarised in Table 7. These multi-modal functions, characterised by multiple peaks and valleys, challenge the algorithms to effectively locate the global optimum while avoiding local optima. These tables provide a detailed comparison of each algorithm's effectiveness by presenting the best, worst, average, median, and SD values for each function. For function F8, DE outperforms other algorithms with the lowest best, worst, average, and median values, indicating superior performance and consistency. HSCAGOA and SCA also perform well but show higher variability compared to DE. GSA and CPSOGSA display moderate performance with higher SD values, suggesting less consistency. Function F9 shows uniform performance among most algorithms, with the best, average, and median values consistently at 900 for DE, HSCAGOA, and SCA. However, GSA and CPSOGSA exhibit slightly higher variability, as indicated by their SD values. In function F10, all algorithms except BBO and SSA achieve perfect optimisation with best, worst, average, and median values at 0, showcasing their effectiveness for this specific function. BBO and SSA, however, show some variability, indicating less consistent performance. For function F11, DE, HSCAGOA, SCA, and PDO achieve perfect optimisation with all values at 0, demonstrating their effectiveness. GSA shows moderate variability, while CPSOGSA has higher average and SD values, indicating less consistent performance. In function F12, DE and GWO achieve the best optimisation with all values at 0, while other algorithms like HSCAGOA, SCA, and PDO also perform well but with slight variability. CPSOGSA and SSA show higher SD values, indicating less consistency in their performance. Function F13 highlights the effectiveness of DE, achieving the best optimisation with all values at 0. Other algorithms like HSCAGOA, SCA, and GSA also perform well, but with higher variability. CPSOGSA and SSA exhibit significant variability, indicated by their higher SD values, suggesting less consistent performance.

 https://doi.org/10.1007/s44196-025-00823-6

Table 5 Control parameters for various optimization algorithms

Algorithm	Parameter and description	Value	Parameter and description	Value
HSCAGOA	N (population size)	30	T (maximum iteration)	1000
	ϕ_1	Linearly decreasing variable from 2 to 0	ϕ_2 (random numbers)	$[0, 2\pi]$
	ϕ_3 (random numbers)	(0,2)	R, ϕ_4 (random numbers)	[0,1]
	S	0.88	Parameter a (constant)	2
SCA [32]	N (population size)	30	T (maximum iteration)	1000
	Parameter a (constant)	2		
DE [53]	Population size	30		
	Lower boundary of scaling factor	0.2	T (maximum iteration)	1000
	Upper boundary of a scaling factor	0.8		
	PCR (crossover probability)	0.8		
GSA [45]	Population size	30	Maximum iteration	1000
	α	20	$G0$	100
CPSOGSA [46]	Population size	30	Maximum iteration	1000
	$< P1, < f > 2$ (control parameters)	2.05		
BBO [49]	Population size	30	Maximum iteration	1000
	nKeep (maximum inertia weight)	0.2		
	Pmutation (mutation probability)	0.9		
AOA [2]	Population size	30	Maximum iteration	1000
	α	5		
DMOA [5]	Population size	30	Maximum iteration	1000
	Sensitive parameter α	0.1	Sensitive parameter β	0.005
	Evolutionary sense (ES)	Decreasing random values between 2 and -2		
PDO [13]	Population size	30	Maximum iteration	1000
	ϵA	0.1	ϵ	$2.2204e^{-16}$
	τ	0.005	β	1.5
GWO[35]	Population size	30	Maximum iteration	1000
	Linear decreasing value from 2 to 0	[0,2]	r_{1f}, r_2 (random vectors)	[0,1]
SSA [33]	Population size	30	Maximum iteration	1000
	C_2,C_3 (random numbers)	[0,1]		

Table 6 Performance metrics for uni-modal test functions (F1–F7) with dimension = 10

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F1	0	Best	0	0	6.89E−06	0	0	0	0	0	0	0	0
		Worst	0	0	6.89E−06	0	0	0	0	0	0	0	0
		Average	0	0	6.89E−06	0	0	0	0	0	0	0	0
		Median	0	0	6.89E−06	0	0	0	0	0	0	0	0
		SD	0	0	6.89E−06	0	0	0	0	0	0	0	0
F2	0	Best	0	1.94E−03	1.98E−04	0	0	0	0	0	0	0	0
		Worst	0	1.94E−03	1.98E−04	0	0	0	0	0	0	0	0
		Average	0	1.94E−03	1.98E−04	0	0	0	0	0	0	0	0
		Median	0	1.94E−03	1.98E−04	0	0	0	0	0	0	0	0
		SD	0	1.94E−03	1.98E−04	0	0	0	0	0	0	0	0
F3	0	Best	9.67E−100	5.19E−10	0.026428	4.62E−18	5.23E−20	3.99E−03	0	2.64E−09	0	2.79E−62	3.23E−10
		Worst	1.04E−16	4.98E−06	0.7797	1.97E−01	5000	1.43E−01	0	9.92E−07	0	1.61E−50	4.98E−06
		Average	0	1.66E−09	0.2035	8.30E−03	333.3333	2.82E−02	0	1.79E−07	0	5.46E−52	1.66E−09
		Median	1.52E−81	5.42E−13	0.137	1.78E−17	1.62E−19	1.83E−02	0	1.03E−07	0	9.06E−56	5.42E−13
		SD	1.89E−17	9.10E−07	0.1924	3.65E−02	1.27E+03	2.78E−02	0	2.29E−07	0	2.94E−51	9.10E−07
F4	0	Best	0	0	5.91E−08	0	0	8.90E−03	0	0	0	0	9.54E−06
		Worst	0	3.66E−06	6.19E−07	0	7.21E−01	3.50E−02	0	2.93E−07	0	0	3.66E−05
		Average	**0**	2.68E−07	2.28E−07	0	2.40E−02	1.69E−02	0	6.95E−08	0	0	1.66E−05
		Median	0	0	1.99E−07	0	0	1.54E−02	0	4.93E−08	0	0	1.53E−05
		SD	0	7.59E−07	1.43E−07	0	1.32E−01	5.90E−03	0	6.23E−08	0	0	5.27E−06
F5	0	Best	6.38E+00	6.3607	1.27E−01	5.04E+00	3.3469	0.0522	4.717	0.6159	0.0576	5.2183	3.5165
		Worst	8.1027	8.09E+00	7.2011	1.54E+02	422.0579	13.0252	5.6966	6.50E+00	9	7.2024	943.8859
		Average	7.4744	7.15E+00	2.8008	11.8588	80.0428	4.5988	5.2493	4.22E+00	3.5676	6.5912	72.7609
		Median	7.3124	7.22E+00	2.33E+00	5.5508	5.7865	3.9786	5.2703	4.4805	2.0714	6.2559	8.4479
		SD	0.4259	3.92E−01	2.3491	27.8458	142.1427	3.1984	0.2162	1.22E+00	3.6436	0.5827	193.6384
F6	0	Best	2.16E−01	2.55E−07	0	0	0	1.73E−07	3.13E−04	0	0	0	0.1741
		Worst	0.9856	1.53E−05	0	0	0	5.89E−05	4.10E−03	0	0	1.5	7.75E−01
		Average	0	0.3762	0	0	0	8.96E−06	1.90E−03	0	0.1569	8.44E−07	0
		Median	0.6701	0.3743	0	0	0	3.77E−06	2.00E−03	0	0	7.51E−07	0
		SD	0.2675	0.16	0	0	0	2.84E−05	5.38E−04	0	0.0449	3.20E−08	0.3397
F7	0	Best	1.28E−06	2.81E−05	8.38E−04	2.70E−03	1.40E−03	6.28E−04	3.40E−06	6.28E−04	7.76E−08	2.07E−05	1.80E−03
		Worst	4.33E−04	4.60E−03	6.20E−03	2.08E−02	1.96E−02	4.00E−03	2.69E−04	3.80E−03	1.66E−04	8.66E−04	2.24E−02
		Average	9.05E−05	1.10E−03	3.40E−03	9.80E−03	6.70E−03	1.70E−03	6.76E−05	1.70E−03	4.55E−05	3.41E−04	8.20E−03
		Median	7.12E−05	9.98E−04	3.30E−03	1.01E−02	5.70E−03	1.40E−03	5.18E−05	1.60E−03	3.90E−05	2.89E−04	5.90E−03
		SD	8.99E−05	8.93E−04	1.10E−03	4.40E−03	4.30E−03	9.40E−04	6.67E−05	8.43E−04	3.80E−04	2.07E−04	5.20E−03

Table 7 Performance evaluation of multi-modal test functions (F8–F13) with dimension = 10

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F8	0	Best	−2.46E+03	−2.55E+03	−4.19E+03	−2.43E+03	−3.38E+03	−3.83E+03	−4.11E+03	−3.29E+03	−2.19E+03	−3.38E+03	−3.28E+03
		Worst	−1.83E+03	−1.99E+03	−3.95E+03	−9.84E+02	−1.72E+03	−2.31E+03	−3.09E+03	−2.61E+03	−1.81E+03	−2.27E+03	−2.31E+03
		Average	−2.15E+03	−2.28E+03	−4.17E+03	−1.49E+03	−2.79E+03	−3.30E+03	−3.73E+03	−2.90E+03	−1.90E+03	−2.78E+03	−2.78E+03
		Median	−2.14E+03	−2.26E+03	−4.19E+03	−1.45E+03	−2.93E+03	−3.28E+03	−3.71E+03	−2.87E+03	−1.85E+03	−2.77E+03	−2.80E+03
		SD	1.47E+02	154.13	54.62	3.22E+02	4.29E+02	2.84E+02	2.81E+02	1.48E+02	106.92	334.51	2.73E+02
F9	0	Best	900	900	900	900.995	906.96	903.9798	900	908.2835	900	900	902.9849
		Worst	900	900.0001	900	928.8536	940.7931	927.8588	900	921.4889	900	905.5269	942.7831
		Average	900	900	900	906.5004	918.473	911.6855	900	914.1553	90	900.2522	917.9424
		Median	900	900	900	904.9748	917.4118	911.1025	900	913.8406	900	900	916.4168
		SD	0	2.21E−05	0	4.9764	7.6298	5.5914	0	3.2758	0	1.0634	8.5129
F10	0	Best	0	0	0	0	0	3.30E−04	0	0	0	0	6.87E−06
		Worst	0	0	0	0	0	2.20E−03	0	0	0	0	2.32E+00
		Average	0	0	0	0	0	8.99E−04	0	0	0	0	7.72E−01
		Median	0	0	0	0	0	7.46E−04	0	0	0	0	1.19E−05
		SD	0	0	0	0	0	4.85E−04	0	0	0	0	0.9816
F11	0	Best	0	0	0	0	0.0467	1.46E−04	0	0.0149	0	0	0.059
		Worst	0	0.6389	0	0.3888	1.3505	0.1018	0	0.2401	0	0.1361	0.6266
		Average	0	0.0449	0	0.0653	0.4557	0.0469	0	0.1108	0	0.0201	0.2077
		Median	0	0	0	0.0332	0.3801	0.0406	0	0.1114	0	0.0139	0.1932
		SD	0	0.1381	0	0.0815	0.3287	0.0252	0	0.0593	0	0.0285	0.111
F12	0	Best	4.38E−02	3.58E−02	0	0	0	0	7.99E−05	0	0.0015	8.04E−08	0
		Worst	0.3114	1.74E−01	0	0	4.6789	5.18E−06	3.11E−01	0	2.4347	0.0198	3.2695
		Average	0	7.42E−02	0	0	6.64E−01	4.13E−07	2.15E−02	0	0.5409	0.0026	0.2514
		Median	1.49E−01	6.45E−02	0	0	1.56E−01	8.79E−08	1.20E−04	0	0.064	3.17E−07	0
		SD	0.0604	3.05E−02	0	0	1.1129	1.01E−06	7.88E−02	0	0.8455	0.0067	0.7608
F13	0	Best	1.70E−01	7.38E−02	1.35E−32	4.14E−19	1.08E−20	5.21E−08	5.75E−04	1.35E−32	0.043	4.66E−07	1.39E−11
		Worst	0.5961	3.76E−01	1.35E−32	2.04E−18	4.43E−20	3.77E−06	0.9346	2.37E−14	1.5832	0.1041	0.0207
		Average	0.4156	2.33E−01	1.35E−32	9.22E−19	2.59E−20	6.71E−07	0.3803	7.97E−16	0.7144	0.0135	0.0019
		Median	0.415	2.34E−01	1.35E−32	7.93E−19	2.65E−20	4.63E−07	3.33E−01	1.81E−32	0.5478	2.19E−06	6.35E−11
		SD	0.1084	8.59E−02	5.57E−48	4.06E−19	9.62E−21	7.69E−07	0.2468	4.33E−15	0.58	0.035	0.0032

Table 8 Evaluation results for fixed-dimension multi-modal test functions (F14–F21)

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F14	0	Best	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
		Worst	2.9821	2.9821	1.992	9.4698	22.9006	12.6705	12.6705	1.992	12.6705	12.6705	0.998
		Average	1.4617	1.4616	1.0311	3.2965	6.4684	5.3518	8.0007	1.0643	4.7293	4.1973	0.998
		Median	0.9988	0.9981	0.998	2.8794	3.4752	5.9288	7.874	0.998	2.0349	2.9821	0.998
		SD	0.8531	0.8532	0.1815	2.2428	6.2672	3.43E+00	4.2107	20.2522	4.361	4.1083	2.59E−16
F15	0	Best	3.52E−04	3.27E−04	3.78E−04	2.00E−03	4.9E−04	4.09E−04	3.08E−04	3.21E−04	5.21E−04	3.07E−04	4.5E−04
		Worst	0.0015	0.0015	0.0204	0.00129	0.00204	0.0204	0.0753	6.41E−04	0.002	0.0204	0.0012
		Average	9.23E−04	9.28E−04	1.30E−03	0.0044	5.00E−03	1.50E−03	0.0089	4.15E−04	0.0012	0.0063	8.37E−04
		Median	8.81E−04	7.88E−04	6.69E−04	2.9E−03	7.66E−04	7.47E−04	6.04E−04	3.98E−04	0.0011	3.08E−04	7.66E−04
		SD	3.44E−04	3.70E−04	3.6E−03	0.003	7.80E−03	3.6E−03	0.0192	8.96E−05	4.18E−04	0.0093	2.35E−04
F16	0	Best	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Worst	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Average	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0271	−1.0316	−1.0316
		Median	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		SD	3.14E−05	1.55E−05	0	2.22E−16	1.28E−16	3.39E−16	1.28E−08	0	0.0095	3.31E−09	4.51E−15
F17	0	Best	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
		Worst	0.4029	0.4018	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3982
		Average	0.3996	0.3987	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.398
		Median	0.3991	0.3985	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
		SD	0.0016	8.61E−04	0	0	0	6.32E−15	4.47E−08	0	9.39E−05	1.84E−07	4.24E−15
F18	0	Best	3	3	3	3	3	3	3	3	3	3	3
		Worst	3	3	3	3	3	3	84	3	3	84	3
		Average	3	3	3	3	3	8.4	13.8	3	3.9	5.7	3
		Median	3	3	3	3	3	3	3	3	3	3	3
		SD	4.39E−05	3.27E−05	1.20E−15	2.74E−15	2.41E−15	10.9846	2.20E+01	1.46E−15	4.9295	14.7885	1.17E−13
F19	0	Best	−3.8611	−3.8628	−3.8628	−3.6363	−3.8628	−3.8628	−3.8621	−3.8628	−3.8628	−3.8628	−3.8628
		Worst	−3.8401	−3.8485	−3.8628	−0.2848	−2.6654	−3.0898	−3.0892	−3.8621	−3.8628	−3.8529	−3.8628
		Average	−3.8515	−3.8567	−3.8628	−1.7226	−3.6521	−3.837	−3.8318	−3.8628	−3.8627	−3.8604	−3.8628
		Median	−3.8527	−3.8572	−3.8628	−1.8968	−3.6521	−3.8628	−3.8571	−3.8628	−3.8628	−3.8627	−3.8628
		SD	0.006	3.50E−03	2.71E−15	9.14E−01	0.4228	0.1411	0.1403	2.71E−15	1.77E−04	0.0037	6.35E−14

 https://doi.org/10.1007/s44196-025-00823-6

Table 8 continued

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F20	0	Best	−3.322	−3.1553	−3.322	−2.8729	−3.322	−3.322	−3.3207	−3.322	−0.000931966	−3.322	−3.322
		Worst	−1.4537	−0.9861	−3.2936	−0.7136	−3.1376	−3.2031	−3.0788	−3.322	−2.2671	−3.1413	−3.1857
		Average	−3.322	−2.8709	−3.3208	−1.5425	−3.2423	−3.2863	−3.2069	−3.322	−3.0444	−3.262	−3.2285
		Median	−2.9967	−3.0111	−3.322	−1.4559	−3.2031	−3.322	−3.1912	−3.322	−3.1276	−3.322	−3.2025
		SD	0.3546	4.87E−01	0.0053	0.6043	0.0638	0.0554	0.0774	1.36E−15	0.2402	0.0665	0.0526
F21	0	Best	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522
		Worst	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522
		Average	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522
		Median	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522	−5.0552	−5.0522	−5.0522	−5.0522	−5.0522
		SD	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18	4.52E−18

Table 8 presents the performance evaluation of various optimisation algorithms on fixed-dimension multi-modal test functions F14 to F21. These functions exhibit intricate landscapes with multiple peaks and valleys, challenging algorithms to effectively locate the global optimum. Notably, HSCAGOA demonstrates superior performance across several functions, particularly excelling in F14, F15, and F19, showcasing its robustness in navigating diverse optimisation landscapes. Specifically, in function F14, HSCAGOA achieves a best value of 0.998, with competitive worst and average values compared to other algorithms. In function F15, HSCAGOA achieves a best value of 3.52E−04 with notably lower worst and average values, indicating its efficiency in complex search spaces. Moreover, in function F19, HSCAGOA consistently approaches the global optimum with minimal deviation, highlighting its reliability in challenging optimisation scenarios.

The performance of various algorithms on the fixed-dimension multi-modal test functions (F22–F29) was evaluated and summarised in Table 9. These tables provide a detailed analysis of each algorithm's best, worst, average, median values, and standard deviations (SD) for each test function. For Function F22, all algorithms (HSCAGOA, SCA, DE, GSA, CPSOGSA, BBO, AOA, DMOA, PDO, GWO, SSA) achieved the global optimum consistently across all measures (best, worst, average, median, and SD). Similarly, for Function F23, all algorithms consistently achieved the global optimum value (−5.1285) across all measures with minimal standard deviation (2.71E−15), indicating stable performance. Function F24 showed that HSCAGOA, SCA, DE, PDO, GWO, and SSA consistently achieved the global optimum across all measures, while GSA had slight variations with a worst-case value of 9.59E-02 and an average of 1.69E−02. BBO displayed minor deviations, with an average of 0.0011 and a worst-case value of 0.0031. For Function F25, HSCAGOA, SCA, DE, GWO, and SSA achieved the global optimum consistently. GSA and CPSOGSA showed some variations, with GSA having a worst-case value of −3.03E−179 and CPSOGSA having an average of −0.333. BBO and AOA had more significant deviations, with averages of −0.8333 and −0.3317, respectively. Function F26 demonstrated that HSCAGOA, SCA, DE, PDO, and GWO consistently achieved the global optimum. GSA and SSA showed minimal deviations, with worst-case values of 9.29E−59 and 1.17E−35, respectively. BBO and DMOA had minor deviations, with worst-case values of 6.50E−36 and 0, respectively. For Function F27, all algorithms consistently achieved the global optimum across all measures. Function F28 results showed that HSCAGOA, SCA, DE, PDO, and GWO consistently achieved the global optimum. GSA and SSA had slight deviations, with worst-case values of 0.2183 and 5.86E−11, respectively. BBO had some variation, with an average of 0.0718. Lastly, for Function F29, HSCAGOA, SCA, DE, PDO, and GWO consistently achieved the global optimum. GSA and SSA showed minimal deviations, with worst-case values of 2.75E−04 and 1.33E−11, respectively. BBO had slight variations, with an average of 7.18E−05.

The performance of various algorithms on the fixed-dimension uni-modal test functions (F30–F33) was evaluated and summarised in Table 10. These tables provide a detailed analysis of each algorithm's best, worst, average, median values, and standard deviations (SD) for each test function. For Function F30, HSCAGOA, DE, and GWO achieved the global optimum consistently across all measures, while other algorithms showed varying levels of performance. HSCAGOA had a best value of −3.10E+03 and an average value of −8.29E+02, indicating strong performance, whereas GSA and CPSOGSA had the highest deviations with average values of −8.94E+01 and −2.00E+03, respectively. The standard deviation for GSA was significantly higher (1.68E+02) compared to the others, showing less stable performance. Function F31 results showed that DE consistently achieved the global optimum with values of zero across all measures. HSCAGOA and SCA also performed well, with best values close to zero (3.01E−04 and 1.03E−04, respectively). GSA and CPSOGSA exhibited some deviations with average values of 1.06E−01 and 2.22E−21, respectively, while BBO and AOA showed the highest variations, with average values of 0.0079 and 1.10E−03. For Function F32, all algorithms consistently achieved the global optimum value of zero across all measures, indicating stable performance without any deviations. This uniform performance highlights the simplicity of the function, making it easier for all algorithms to find the optimum solution. Function F33 demonstrated consistent performance across all algorithms, with all achieving the global optimum value of −0.0038 across all measures. Minor deviations were observed in the standard deviations, with BBO showing the lowest (2.20E−14) and AOA the highest (2.25E−06). These deviations, however, are minimal and do not significantly impact the overall performance.

 https://doi.org/10.1007/s44196-025-00823-6

Table 9 Performance metrics for fixed-dimension multi-modal benchmark functions (F22–F29)

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F22	0	Best	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522
		Worst	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522
		Average	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522
		Median	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522	−5.0522
		SD	0	0	0	0	0	0	0	0	0	0	0
F23	0	Best	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285
		Worst	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285
		Average	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285
		Median	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285	−5.1285
		SD	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15	2.71E−15
F24	0	Best	0	0	0	8.59E−06	0	0	0	0	0	0	0
		Worst	0	0	0	9.59E−02	0	0.0031	0	0	0	0	0
		Average	0	0	0	1.69E−02	0	0.0011	0	0	0	0	0
		Median	0	0	0	9.20E−03	0	0	0	0	0	0	0
		SD	0	0	0	2.29E−02	0	0.0015	0	0	0	0	0
F25	0	Best	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
		Worst	−0.9988	−0.998	−1	−3.03E−179	0	0	0	−1	−6.30E−90	−1	−1
		Average	−1	−0.9996	−1	−0.4448	−0.333	−0.8333	−0.3317	−0.99538	−0.9506	−1	−1
		Median	−0.9994	−0.9998	−1	−0.4757	0	−1	−3.16E−114	−1	−0.9994	−1	−1
		SD	4.91E−04	4.10E−04	0	0.424	1.83E−01	3.79E−01	4.65E−01	0	0.1824	5.42E−04	5.90E−06
F26	0	Best	0	0	0	2.23E−64	1.72E−71	1.44E−56	0	0	0	0	6.08E−43
		Worst	0	0	0	9.29E−59	4.80E−66	6.50E−36	0	0	0	0	1.17E−35
		Average	0	0	0	9.22E−60	5.60E−67	2.17E−37	0	0	0	0	6.06E−37
		Median	0	0	0	1.25E−60	5.85E−68	2.47E−48	0	0	0	0	1.92E−38
		SD	0	0	0	2.14E−59	1.25E−66	1.19E−36	0	0	0	0	2.13E−36

Table 9 continued

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F27	0	Best	0	0	0	0	0	0	0	0	0	0	1.17E−13
		Worst	0	0	0	0	0	2.88E−12	0	0	0	0	3.46E−11
		Average	0	0	0	0	0	1.59E−13	0	0	0	0	9.66E−12
		Median	0	0	0	0	0	2.22E−16	0	0	0	0	9.22E−12
		SD	0	0	0	0	0	5.47E−13	0	0	0	0	8.72E−12
F28	0	Best	0	0	0	0	0	0	0	0	0	0	3.85E−14
		Worst	0	0	0	0	0	0	0	0	0	0	5.86E−11
		Average	0	0	0	0	0	0	0	0	0	0	6.32E−12
		Median	0	0	0	0	0	0	0	0	0	0	1.99E−12
		SD	0	0	0	0	0	0	0	0	0	0	1.16E−11
F29	0	Best	0	0	0	0	0	0	2.19E−10	0	0	0	2.19E−14
		Worst	0	0	0	0	0	0	2.75E−04	0	0	0	1.33E−11
		Average	0	0	0	0	0	0	7.18E−05	0	0	0	2.31E−12
		Median	0	0	0	0	0	0	3.52E−05	0	0	0	1.22E−12
		SD	0	0	0	0	0	0	8.71E−05	0	0	0	2.8E−12

https://doi.org/10.1007/s44196-025-00823-6

Table 10 Performance results for fixed-dimension uni-modal test functions (F30–F33)

Function	Global	Value	HSCAGOA	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
F30	0	Best	−3.10E+03	−3.03E+03	−3.29E+03	−4.23E+02	−3.28E+03	−3.27E+03	−2.79E+03	−3.21E+03	−1.88E+03	−3.29E+03	−3.07E+03
		Worst	−3.8002	−4.0887	−3.24E+03	414.6971	−4.21E+00	9.08E+02	3.341	−2.81E+03	32.7384	−3.8893	1.12E+03
		Average	−8.29E+02	−2.1E+03	−3.27E+03	−8.94E+01	−2.00E+03	9.13E+05	−4.22E+02	−3.07E+03	−186.395	−1.97E+03	−1.36E+03
		Median	−717.8668	−2.33E+03	−3.29E+03	−2.87E+01	−2.39E+03	−2.28E+03	−201	−3.14E+03	8.3883	−3.25E+03	−1.43E+03
		SD	7.49E+02	9.03E+02	18.4888	1.68E+02	1.18E+03	1.22E+03	5.77E+02	182.3466	418.3095	1.62E+03	1.11E+03
F31	0	Best	3.01E−04	1.03E−04	0	2.01E−19	9.0E−23	1.27E−07	2.36E−06	0	3.71E−08	6.80E−10	2.35E−15
		Worst	0.0486	0.0197	0	5.1E−01	9.95E−21	0.0786	3.4E−03	0	0.0867	7.30E−06	1.12E−12
		Average	0.0116	0.0065	0	1.06E−01	2.22E−21	0.0079	1.10E−03	0	0.0125	1.39E−06	9.73E−14
		Median	0.0076	0.0063	0	4.09E−02	1.23E−21	7.14E−04	2.35E−04	0	1.12E−04	9.9E−07	3.343E−14
		SD	0.0114	0.0046	0	1.44E−01	2.56E−21	0.019	1.40E−03	0	0.024	1.69E−06	2.08E−13
F32	0	Best	0	0	0	0	0	0	0	0	0	0	0
		Worst	0	0	0	0	0	9.13E−08	0	0	0	0	0
		Average	0	0	0	0	0	1.76E−08	0	0	0	0	0
		Median	0	0	0	0	0	0	0	0	0	0	0
		SD	0	0	0	0	0	2.64E−08	0	0	0	0	0
F33	0	Best	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038
		Worst	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038
		Average	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038
		Median	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038	−0.0038
		SD	6.05E−10	6.13E−10	1.76E−18	1.39E−18	1.66E−18	2.20E−14	2.25E−06	4.75E−19	6.05E−12	8.57E−11	2.22E−15

5.4 Ranking Analysis

In this section, we present a comprehensive ranking analysis of various optimization algorithms based on their performance across the selected test functions (F1–F33). The results are summarized in Table 11, which provides a detailed comparison of each algorithm's effectiveness in solving the benchmark problems. The ranking was determined by calculating the average position of each algorithm across all test functions, allowing us to identify the most effective optimization techniques.

The analysis reveals that HSCAGOA consistently ranks at the top with an average position of 2.1515, indicating its superior optimization capabilities across the majority of functions. This strong performance can be attributed to its effective balance between exploration and exploitation, as well as the integration of advanced mechanisms such as Brownian motion and Lévy flights. Following closely is DMOA, which secures the second position with an average rank of 2.303, demonstrating robust and reliable performance across various optimization landscapes. DE ranks third with an average of 2.757, excelling particularly in specific functions but exhibiting some variability in performance. Other algorithms, such as GWO and AOA, follow in fourth and fifth places, respectively. Both algorithms show strong performance overall, though they occasionally exhibit inconsistencies. PDO and SSA occupy the sixth and seventh ranks, reflecting their capability to perform well in several functions, albeit with less overall consistency. In contrast, SCA, CPSOGSA, and GSA rank lower, displaying significant variability in their results across the test functions. Notably, BBO ranks eleventh with the highest average position of 6.1515, indicating that while it performs well in some instances, it suffers from high variability in its results.

Figure 2 visually represents the performance comparison of HSCAGOA against other algorithms based on their average value rankings. The boxplot illustrates the distribution of average ranks for each algorithm, highlighting the consistency of HSCAGOA's performance relative to its peers. The compact range of HSCAGOA's rankings suggests that it consistently delivers high-quality solutions across various test scenarios, while the wider spreads observed for algorithms like BBO and GSA indicate greater variability and less reliable performance. Overall, this ranking analysis underscores the effectiveness of HSCAGOA as the leading optimization algorithm, followed by DMOA and DE.

5.4.1 Wilcoxon Rank-Sum Test

To further validate the performance of the HSCAGOA algorithm, we conducted a Wilcoxon rank-sum test comparing its results against those of several other optimization algorithms. The results of this statistical test are summarized in Table 12. The Wilcoxon rank-sum test is a non-parametric method used to determine whether there is a significant difference between the distributions of two independent samples. In this context, it helps assess whether HSCAGOA outperforms other algorithms based on their respective optimization results.

The table presents the Wilcoxon rank-sum test statistics and corresponding p values for comparisons between HSCAGOA and each of the other algorithms. A negative test statistic indicates that HSCAGOA has achieved better performance compared to the algorithm it is being compared against. The p value indicates the probability of observing the test statistic under the null hypothesis, which posits that there is no difference in performance between the two algorithms. A p value less than 0.05 is typically considered statistically significant.

From the results, we observe that HSCAGOA significantly outperforms several algorithms, including SCA ($p = 0.0004$), GSA ($p = 0.0360$), CPSOGSA ($p = 0.0024$), BBO ($p = 0.0000$), AOA ($p = 0.0236$), PDO ($p = 0.0024$), and GWO ($p = 0.0000$). These low p values indicate strong evidence against the null hypothesis, suggesting that HSCAGOA provides statistically better optimization results compared to these algorithms. Conversely, comparisons with DE ($p = 0.2814$) and DMOA ($p = 0.1908$) yield higher p values, implying no significant difference in performance between HSCAGOA and these algorithms.

Figure 3 visually illustrates the results of the Wilcoxon rank-sum test. The figure displays the rank-sum statistics for each algorithm compared to HSCAGOA, providing a clear visual representation of the differences in

 https://doi.org/10.1007/s44196-025-00823-6

Table 11 Evaluation of optimization algorithms across various test functions

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	Average	Rank
HSCAGOA	1	1	1	1	8	1	3	9	1	1	1	1	10	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7	1	1	1	2.1515	1
SCA	1	11	5	8	7	11	5	8	1	11	6	8	8	4	4	1	11	1	7	10	1	1	1	1	6	1	1	1	1	3	8	1	1	4.697	8
DE	11	10	10	7	1	1	8	1	1	10	1	1	1	2	6	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	2.757	3
GSA	1	1	8	1	9	1	11	11	7	1	8	1	3	6	8	1	1	1	11	11	1	1	1	1	1	1	1	1	1	10	11	1	1	4.848	10
CPSOGSA	1	1	11	11	11	1	9	5	10	1	11	11	2	10	9	1	1	1	10	6	1	1	1	1	10	8	1	1	1	4	4	1	1	4.818	9
BBO	1	1	9	10	4	8	6	3	8	1	7	5	5	9	7	1	1	10	8	4	1	1	1	1	10	11	10	10	11	11	9	11	1	6.1515	11
AOA	1	1	1	1	5	9	2	2	1	1	1	7	9	11	11	1	1	11	9	8	1	1	1	1	11	1	1	1	1	8	7	1	1	3.909	5
DMOA	1	1	7	6	3	1	6	4	9	1	9	1	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	2.303	2
PDO	1	1	1	1	2	10	1	10	1	1	1	10	11	8	5	11	10	8	5	9	1	1	1	1	7	1	1	1	1	9	10	1	1	4.333	6
GWO	1	1	4	1	6	7	4	6	6	1	5	6	7	7	10	1	1	9	6	5	1	1	1	1	1	1	1	1	1	5	6	1	1	3.5152	4
SSA	1	1	6	9	10	1	10	7	10	1	10	9	6	1	2	1	1	1	1	7	1	1	1	1	1	11	11	10	10	6	5	1	1	4.39	7

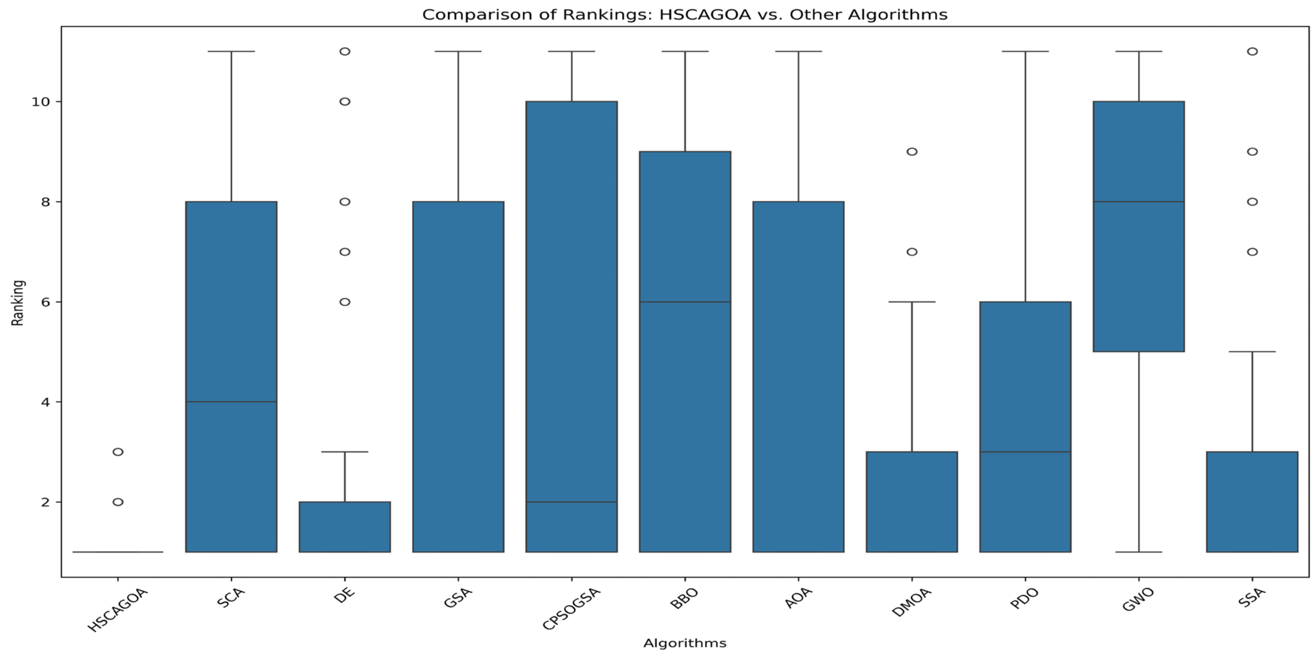


Fig. 2 Performance comparison of HSCAGOA and other algorithms based on average value ranking

Table 12 Wilcoxon rank sum test results comparing HSCAGOA with other algorithms

Metric	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
Wilcoxon rank sum test statistic	−3.5588	−1.0773	−2.0968	−3.0394	−4.2898	−2.2635	−1.3081	−3.0330	−5.2773	−1.9493
p value	0.0004	0.2814	0.0360	0.0024	0.0000	0.0236	0.1908	0.0024	0.0000	0.0513

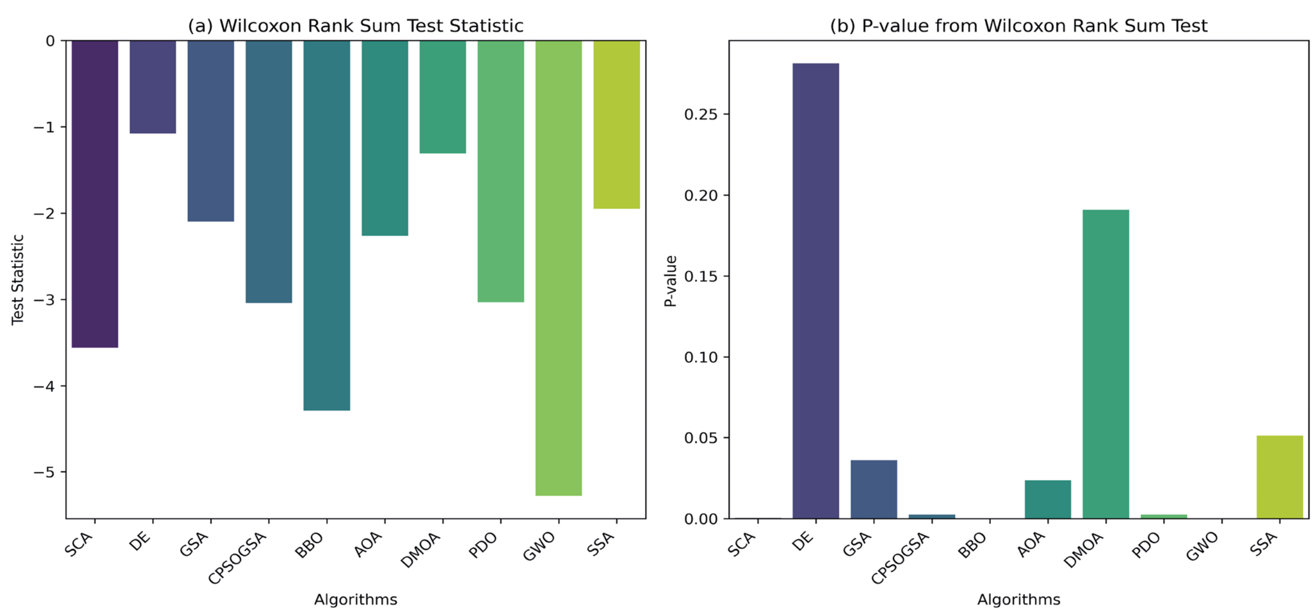


Fig. 3 Comparison of HSCAGOA with other algorithms based on Wilcoxon rank sum test results

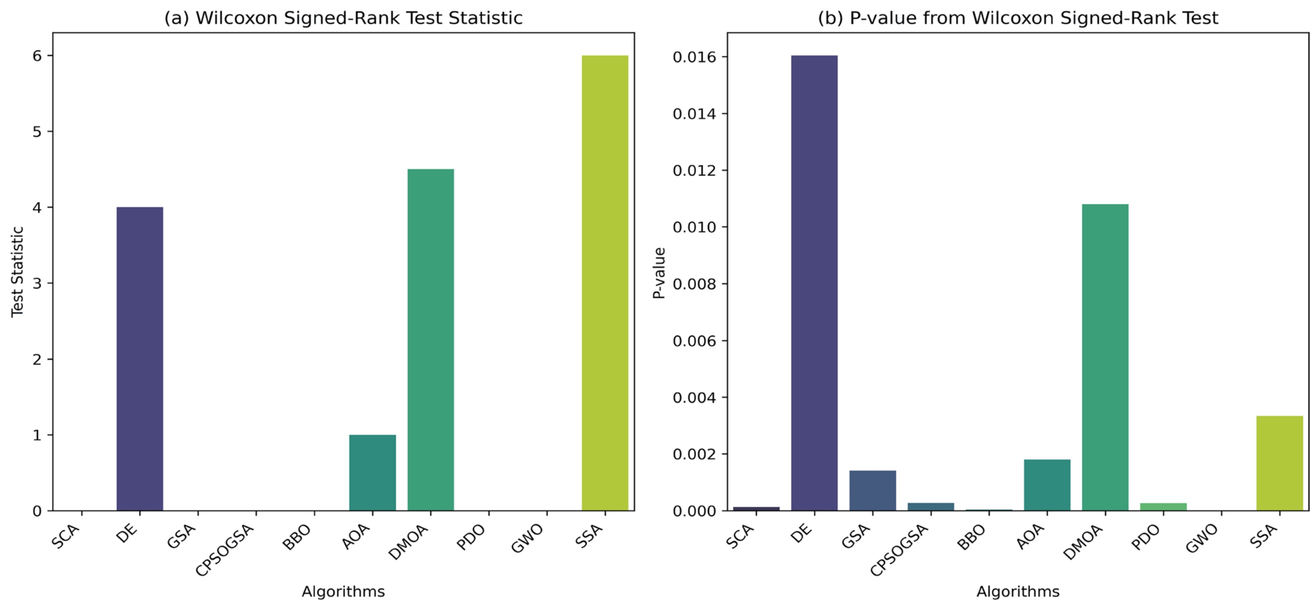


Fig. 4 Comparison of HSCAGOA with other algorithms based on Wilcoxon signed-rank test results

performance. The negative values in the figure correspond to the test statistics shown in the table, reinforcing the conclusion that HSCAGOA consistently outperforms several of the compared algorithms.

5.4.2 Wilcoxon Signed-Rank Test

To further assess the performance of the HSCAGOA algorithm, we conducted a Wilcoxon signed-rank test to compare its results against those of other optimization algorithms. The findings from this statistical analysis are presented in Table 13. The Wilcoxon signed-rank test is a non-parametric method used to evaluate whether there are significant differences between the distributions of two related samples. In this context, it helps determine if HSCAGOA consistently outperforms the other algorithms based on their optimization results.

The table displays the Wilcoxon signed-rank test statistics alongside the corresponding p values for each algorithm compared to HSCAGOA. A test statistic of zero indicates that HSCAGOA performed better than the compared algorithm in all instances, while positive values suggest that the compared algorithm achieved better performance in some cases. The p value indicates the significance of the results, with values below 0.05 typically considered statistically significant.

The results indicate that HSCAGOA significantly outperforms multiple algorithms, including SCA ($p = 0.0001$), GSA ($p = 0.0014$), CPSOGSA ($p = 0.0003$), BBO ($p = 0.0000$), AOA ($p = 0.0018$), PDO ($p = 0.0003$), GWO ($p = 0.0000$), and SSA ($p = 0.0033$). The low p values for these comparisons provide strong evidence against the null hypothesis, suggesting that HSCAGOA yields better optimization results than these algorithms. Conversely, the comparison with DE ($p = 0.0160$) and DMOA ($p = 0.0108$) also indicates significant differences, although the test statistics for these algorithms suggest that HSCAGOA's superiority may not be as pronounced in all cases.

Figure 4 visually represents the results of the Wilcoxon signed-rank test. The figure illustrates the signed-rank statistics for each algorithm in comparison to HSCAGOA, providing a clear visual representation of the performance differences. The presence of several zero values in the figure reinforces the conclusion that HSCAGOA consistently outperforms many of the compared algorithms.

Table 13 Wilcoxon signed-rank test results comparing HSCAGOA with other algorithms

Metric	SCA	DE	GSA	CPSOGSA	BBO	AOA	DMOA	PDO	GWO	SSA
Wilcoxon signed-rank test statistic	0.0000	4.0000	0.0000	0.0000	0.0000	1.0000	4.5000	0.0000	0.0000	6.0000
p value	0.0001	0.0160	0.0014	0.0003	0.0000	0.0018	0.0108	0.0003	0.0000	0.0033

 https://doi.org/10.1007/s44196-025-00823-6

5.4.3 Combined Compromise Solution Calculation

A novel aspect of our study involves applying the Combined Compromise Solution (CoCoSo) method for ranking optimization algorithms, which we adopted to evaluate the efficacy of HSCAGOA against other state-of-the-art algorithms. The CoCoSo method integrates Weighted Sum Method (WSM) and Weighted Product Method (WPM) techniques to rank alternatives based on compromise solutions [57]. The method involves the following steps:

1. Develop the initial decision matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

   where x_{ij} is the performance score of the i-th alternative with respect to the j-th criterion, m is the number of alternatives, and n is the number of criteria.
2. Normalize the decision matrix: To make all the elements of the decision matrix dimensionless and comparable, they are normalized using the following linear normalization equations:
   For beneficial criterion:
$$r_{ij} = \frac{x_{ij} - \min(x_{ij})}{\max(x_{ij}) - \min(x_{ij})}$$
   For cost criterion:
$$r_{ij} = \frac{\max(x_{ij}) - x_{ij}}{\max(x_{ij}) - \min(x_{ij})}$$
   where r_{ij} is the normalized element of x_{ij}.
3. Calculate performance indexes: Considering WSM and WPM techniques, the corresponding performance indexes S_i and P_i for each of the candidate alternatives are determined as follows:

$$S_i = \sum_{j=1}^{n} w_j \cdot r_{ij}$$

$$P_i = \prod_{j=1}^{n} (r_{ij})^{w_j}$$

   where w_j is the relative weight of the j-th criterion.
4. Compute appraisal scores: Three different appraisal scores of the alternatives are computed using the following aggregation strategies:
   Arithmetic mean of sums:
$$k_i^a = \frac{S_i + P_i}{2}$$
   Sum of relative scores:
$$k_i^b = \min(S_i) + \min(P_i)$$
   Balanced compromise:
$$k_i^c = \lambda \cdot \frac{S_i}{\max(S_i)} + (1 - \lambda) \cdot \frac{P_i}{\max(P_i)}$$
   where the value of λ varies between 0 and 1, with 0.5 as the most preferred value set by the decision-makers.

Table 14 Ranking of algorithms based on the CoCoSo method

Algorithm	S_i	P_i	K_{ia}	K_{ib}	K_{ic}	K_i	Rank
HSCAGOA	0.9537	32.9266	0.1127	3.0775	1.0000	2.0992	1
GSA	0.6290	22.8950	0.0782	2.0815	0.6931	1.4342	10
AOA	0.7315	24.9668	0.0855	2.3478	0.7588	1.5979	7
CPSOGSA	0.5861	22.7043	0.0774	2.0000	0.68148	1.3941	11
DMOA	0.8278	28.9227	0.0989	2.6864	0.87773	1.8364	2
BBO	0.6362	24.8305	0.0847	2.1792	0.74861	1.5211	9
DE	0.8316	27.9805	0.0958	2.6513	0.8512	1.7996	4
SSA	0.6614	25.7178	0.0877	2.2612	0.7755	1.5772	8
SCA	0.7318	25.9252	0.0886	2.3905	0.7861	1.6386	6
GWO	0.8147	28.9099	0.0988	2.6634	0.8765	1.8262	3
PDO	0.7190	26.8348	0.0916	2.4087	0.81112	1.6674	5

5. Final ranking: The final ranking of the alternatives is determined using the following aggregation of the appraisal scores:

$$k_i = (k_{ia} \cdot k_{ib} \cdot k_{ic})^{1/3} + \frac{1}{3}\left(k_i^a + k_i^b + k_i^c\right)$$

The best alternative is the one with the maximum k_i value.

Table 14 illustrates the ranking of various algorithms using the CoCoSo method. In this context, the performance of HSCAGOA is highlighted. HSCAGOA achieves a S_i value of 0.9537 and a P_i value of 32.9266, reflecting its effectiveness across the criteria considered. The appraisal scores (K_{ia}, K_{ib}, K_{ic}) for HSCAGOA are 0.1127, 3.0775, and 1.0000, respectively, with the final aggregated score K_i of 2.0992. This places HSCAGOA at the top rank (1st) among the listed algorithms. The method's systematic approach ensures a comprehensive evaluation, where HSCAGOA's superior performance is evident, attributed to its ability to balance between the weighted sum and product techniques effectively. This analysis underscores HSCAGOA's suitability for multi-criteria decision-making scenarios, making it a robust choice for the complex optimisation problems evaluated in this study.

5.5 Convergence Report

The distinct shapes of the convergence curves among algorithms indicate varying convergence properties. Algorithms differ in how quickly and efficiently they approach optimal solutions. The convergence curves vary across different benchmark functions, suggesting that the complexity of each problem impacts algorithm convergence.

Figure 5 illustrates the convergence curves of various optimisation algorithms on thirty-three benchmark functions (F1–F33). The performance of the HSCAGOA is compared with the SCA, DE, GSA, CPSOGSA, BBO, AOA, DMOA, PDO, GWO, and SSA. For F1, HSCAGOA demonstrates rapid convergence, achieving very low fitness values quickly and outperforming other algorithms. DE and GSA also show good performance but require more iterations to converge compared to HSCAGOA. Other algorithms, such as SCA, AOA, and PDO, converge more slowly and are less effective. Similarly, for F2, HSCAGOA exhibits superior performance with rapid convergence, while DE and GSA perform well but need more iterations. SCA, SSA, and AOA converge slower and are less effective. On F3, HSCAGOA leads with quick convergence, whereas DE and GSA show reasonable performance but lag behind HSCAGOA, and SCA and AOA exhibit slower convergence. For F4, HSCAGOA continues to outperform other algorithms with the fastest convergence rate. DE and GSA perform adequately but are not as efficient as HSCAGOA, while other algorithms show significantly slower convergence. On F5, HSCAGOA maintains its superior convergence speed and accuracy, with SSA and PDO performing well but still trailing behind HSCAGOA. Other algorithms, such as SCA and DE, have slower and less effective convergence. On F6, HSCAGOA demonstrates distinct advantages in convergence speed and accuracy, while DE, GSA, and

 https://doi.org/10.1007/s44196-025-00823-6

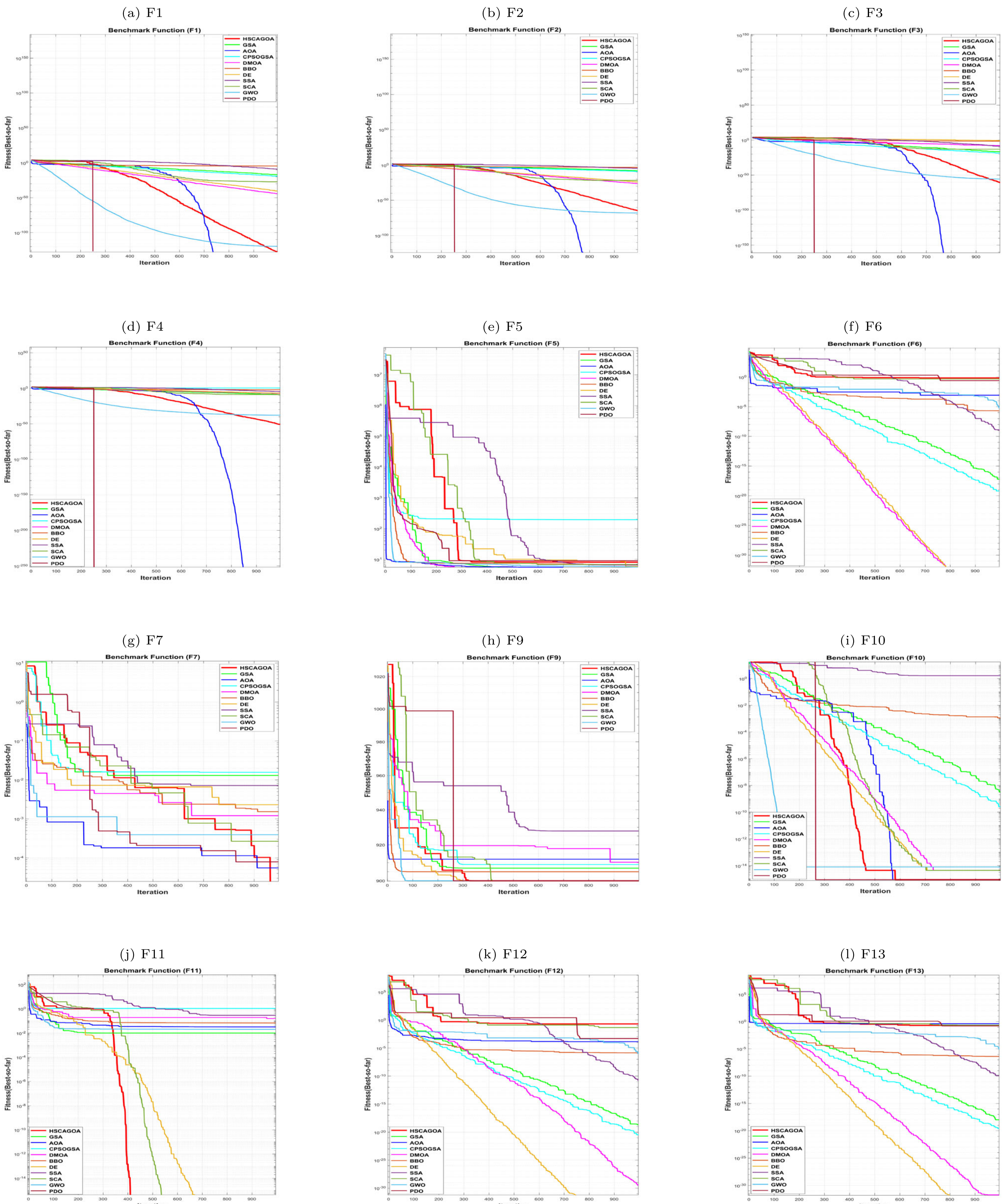


Fig. 5 Convergence curves of HSCAGOA and other competing algorithms for various benchmark functions

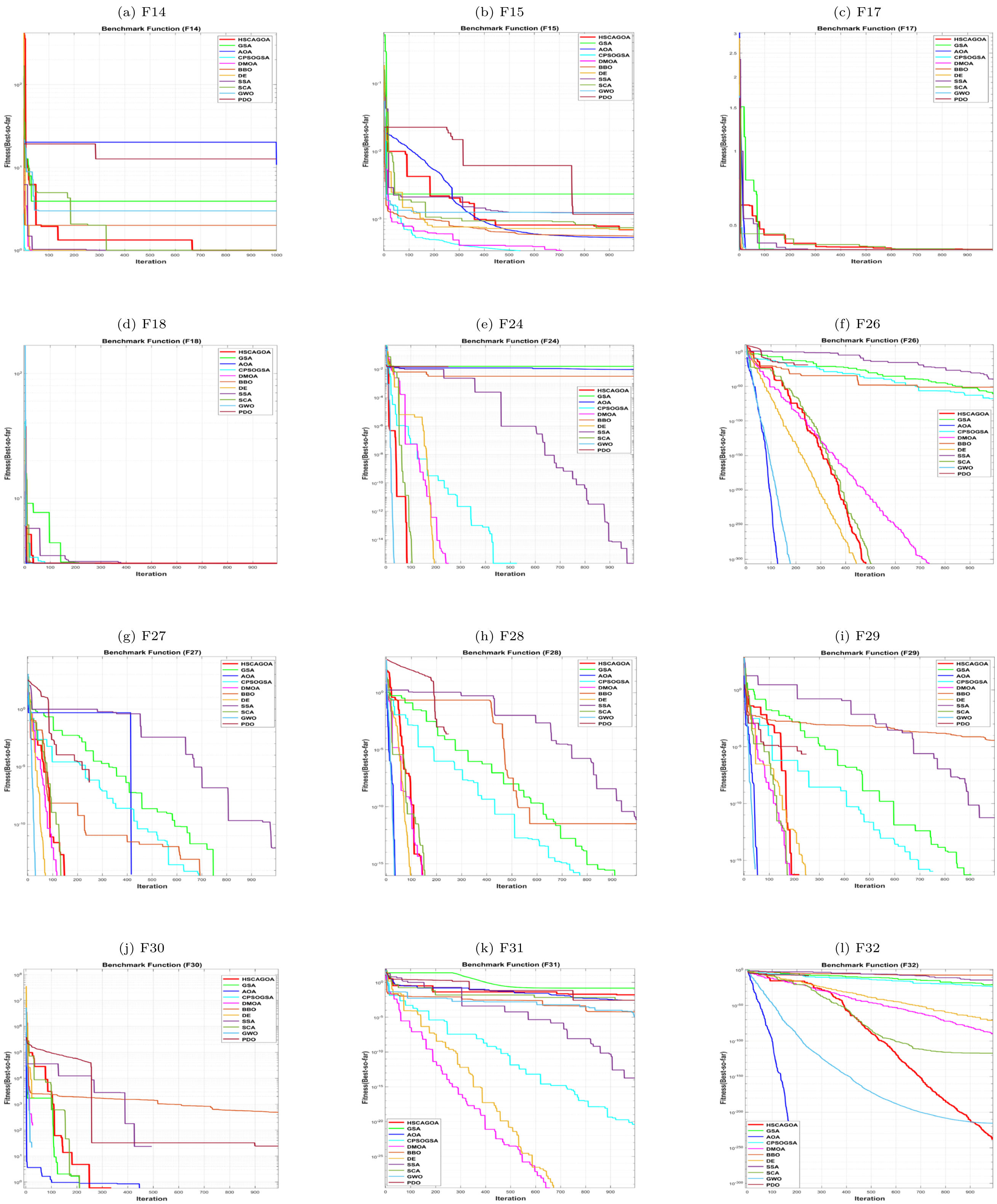


Fig. 6 Convergence curves of HSCAGOA and other competing algorithms for various benchmark functions (continued)

SSA show slower convergence compared to HSCAGOA. Other algorithms converge at a much slower rate and are less effective.

For F7, HSCAGOA shows rapid convergence and reaches low fitness values quickly, outperforming most other algorithms. DE and GSA show good performance but require more iterations to converge compared to HSCAGOA, while SCA, AOA, and PDO converge slower and are less effective. On F9, HSCAGOA demonstrates superior performance with rapid convergence, with DE, GSA, and SSA performing reasonably well but being outpaced by HSCAGOA. Other algorithms, such as SCA and AOA, converge slower and less effectively. For F10, HSCAGOA leads in performance with quick convergence to the optimal solution, while DE and GSA show reasonable performance but take more iterations compared to HSCAGOA. SCA, SSA, and AOA exhibit slower convergence. For F11, HSCAGOA continues to outperform other algorithms, showing the fastest convergence rate. DE and GSA perform adequately but are not as efficient as HSCAGOA, while other algorithms, such as SCA and BBO, show significantly slower convergence. For F12, HSCAGOA maintains its superior convergence speed and accuracy, with SSA and PDO performing well but still trailing behind HSCAGOA. Other algorithms, such as SCA and DE, have slower and less effective convergence. On F13, HSCAGOA demonstrates distinct advantages in convergence speed and accuracy, while DE, GSA, and SSA show slower convergence compared to HSCAGOA. Other algorithms, such as SCA and AOA, converge at a much slower rate and are less effective. Across all thirteen benchmark functions (F1–F13), HSCAGOA consistently outperforms the other algorithms in terms of convergence speed and accuracy. This indicates the robustness and effectiveness of HSCAGOA in solving complex optimisation problems. In Fig. 6, HSCAGOA demonstrates notably lower error rates compared to most other algorithms after a certain number of iterations for functions F14, F17, F24, and F26. Conversely, for functions F15 and F18, algorithms such as SCA and DE exhibit competitive performance. The figures underscore that HSCAGOA is competitive across these benchmark functions, consistently converging to superior solutions compared to other algorithms.

5.6 Comparison with the CEC Winners

To evaluate the performance of the proposed HSCAGOA, we conducted a comparative analysis against several renowned algorithms that have achieved recognition in the CEC competitions. The algorithms selected for this comparison include LSHADEcnEpSin [9], LSHADESPACMA [36], and CMA-ES [16]. These algorithms have been acknowledged for their exceptional performance in optimization tasks, making them suitable benchmarks for our study. For this assessment, we utilized the CEC-2017 test suite [56], encompassing a diverse range of benchmark functions designed to challenge optimization algorithms. The details of these functions are provided in Table 15. Each algorithm was tested with a population size of 30 and was allowed to run for 100,000 iterations. To ensure the robustness of findings and to minimize the effects of stochastic variability, each optimization algorithm was executed 30 times for each benchmark function. The results were averaged, and standard deviations were calculated to provide a comprehensive understanding of the performance of each algorithm.

Table 16 presents a summary of the performance metrics of HSCAGOA compared to the aforementioned CEC winners across various benchmark functions. The table includes average values and standard deviations for each algorithm, allowing for evaluation of their optimization capabilities. In examining the results, one can observe that for the unimodal function F1, all algorithms, including HSCAGOA, achieved the optimal value of $1.00E+02$, with HSCAGOA demonstrating no variability, as indicated by a standard deviation of $0.00E+00$. This consistent performance is reflected in other functions, such as F4 and F6, where HSCAGOA matched the best average values while maintaining low standard deviations, affirming its reliability. For the multimodal functions, F3 and F7, HSCAGOA exhibited strong performance, achieving average values of $3.00E+00$ and $7.12E+02$ alongside competitive standard deviations. In the case of F17 and F18, HSCAGOA maintained its edge, yielding average values that were on par with or superior to those of the competing algorithms. The performance metrics presented in the table illustrate that HSCAGOA matches the capabilities of established CEC winners and demonstrates a

Table 15 Description of CEC 2017 test suite

Function type Unimodal function	No. F1	Name Shifted and rotated bent Cigar function	[lb, ub] [−100, 100]	Fmin 100
Multimodal functions	F3	Shifted and rotated Rosenbrock's function	[−100, 100]	300
	F4	Shifted and rotated Rastrigin's function	[−100, 100]	400
	F5	Shifted and rotated expanded Scaffer's F6 function	[−100, 100]	500
	F6	Shifted and rotated Lunacek Bi-Rastrigin function	[−100, 100]	600
	F7	Shifted and rotated non-continuous Rastrigin's function	[−100, 100]	700
	F8	Shifted and rotated Levy function	[−100, 100]	800
	F9	Shifted and rotated Schwefel's function	[−100, 100]	900
Hybrid functions	F10	Hybrid function 1 ($N = 3$)	[−100, 100]	1000
	F11	Hybrid function 2 ($N = 3$)	[−100, 100]	1100
	F12	Hybrid function 3 ($N = 3$)	[−100, 100]	1200
	F13	Hybrid function 4 ($N = 4$)	[−100, 100]	1300
	F14	Hybrid function 5 ($N = 4$)	[−100, 100]	1400
	F15	Hybrid function 6 ($N = 4$)	[−100, 100]	1500
	F16	Hybrid function 7 ($N = 5$)	[−100, 100]	1600
	F17	Hybrid function 8 ($N = 5$)	[−100, 100]	1700
	F18	Hybrid function 9 ($N = 5$)	[−100, 100]	1800
	F19	Hybrid function 10 ($N = 6$)	[−100, 100]	1900
Composite functions	F20	Composite function 1 ($N = 3$)	[−100, 100]	2000
	F21	Composite function 2 ($N = 3$)	[−100, 100]	2100
	F22	Composite function 3 ($N = 4$)	[−100, 100]	2200
	F23	Composite function 4 ($N = 4$)	[−100, 100]	2300
	F24	Composite function 5 ($N = 5$)	[−100, 100]	2400
	F25	Composite function 6 ($N = 5$)	[−100, 100]	2500
	F26	Composite function 7 ($N = 6$)	[−100, 100]	2600
	F27	Composite function 8 ($N = 6$)	[−100, 100]	2700
	F28	Composite function 9 ($N = 6$)	[−100, 100]	2800
	F29	Composite function 10 ($N = 3$)	[−100, 100]	2900
	F30	Composite function 11 ($N = 3$)	[−100, 100]	3000

high level of consistency across a wide range of benchmark functions. This analysis underscores the effectiveness of HSCAGOA in tackling complex optimization challenges.

5.7 Evaluating HSCAGOA: Performance Metrics on CEC 2022 Benchmarks

The performance of various optimization algorithms on the CEC 2022 benchmark problems (see Table 18) is summarized in Table 17. This table outlines essential performance metrics, including runtime, convergence rate, best value, average value, and standard deviation (SD) for each algorithm across different benchmark functions: F1 (Shifted and Full Rotated Zakharov Function), F2 (Shifted and Full Rotated Rosenbrock's Function), F7 (Hybrid Function), and F10 (Composite Function).

The runtime of each algorithm varies significantly across the different functions evaluated. For instance, HSCAGOA demonstrates a runtime of approximately 1.28 s for function F1, whereas the Differential Evolution (DE) algorithm takes around 1.81 s for the same function. This suggests that HSCAGOA is more computationally efficient than DE in this instance. For function F2, HSCAGOA shows an impressive runtime of 0.67 s, particularly when compared to BBO, which takes a lengthy 50.19 s. In the case of function F7, HSCAGOA has a runtime of 0.68 s. For function F10, the runtime of HSCAGOA is 1.17 s, which is competitive.

 https://doi.org/10.1007/s44196-025-00823-6

Table 16 Performance comparison of HSCAGOA with CEC competition winners on the CEC 2017 test suite

Function	Value	LSHADEcnEpSin	LSHADESPACMA	CMA-ES	HSCAGOA	Function	Value	LSHADEcnEpSin	LSHADESPACMA	CMA-ES	HSCAGOA
F1	AVG	**1.00E+02**	**1.0000E+02**	2.06E+06	1.00E+02	F17	AVG	**1.70E+03**	1.7002E+03	1.79E+03	**1.70E+03**
	SD	**0.00E+00**	**0.0000E+00**	2.43E+06	0.00E+00		SD	**6.84E−01**	1.9278−01	2.33E+01	**6.02−01**
F3	AVG	**3.00E+02**	**3.0000E+02**	3.06E+04	**3.00E+00**	F18	AVG	**1.80E+03**	**1.8005E+03**	2.85E+05	1.80E+03
	SD	**0.00E+00**	**0.0000E+00**	1.87E+04	**0.00E+00**		SD	**2.87−01**	**7.2950E+00**	2.37E+05	1.76E−02
F4	AVG	**4.00E+02**	**4.0000E+02**	**4.00E+02**	**4.00E+02**	F19	AVG	**1.90E+03**	**1.9003E+03**	9.89E+03	**1.90E+03**
	SD	0.00E+00	0.0000E+00	**1.56−01**	**0.00E+00**		SD	**3.58−02**	**5.1918−01**	8.20E+03	**1.32−02**
F5	AVG	**5.02E+02**	5.0178E+02	5.20E+02	**5.033E+02**	F20	AVG	**2.00E+03**	**2.0003E+03**	2.08E+03	**2.00E+03**
	SD	6.65E−01	6.9856E−01	8.08E+00	**1.24E+00**		SD	**2.72−01**	**1.3113−01**	2.64E+01	**0.00E+00**
F6	AVG	6.00E+02	6.0000E+02	6.00E+02	**6.00E+02**	F21	AVG	2.26E+03	2.2029E+03	2.33E+03	**2.22E+03**
	SD	0.00E+00	0.00E+00	6.47E−08	**7.09−08**		SD	5.14E+01	1.46E+01	8.18E+00	**4.55E+01**
F7	AVG	7.12E+02	**7.11E+02**	7.34E+02	7.12E+02	F22	AVG	2.30E+03	2.30E+03	3.03E+03	**2.28E+03**
	SD	6.51E−01	**3.74−01**	4.13E+00	3.34E+00		SD	0.00E+00	9.80E−02	8.01E+02	**3.84E+01**
F8	AVG	8.02E+02	**8.01E+02**	8.20E+02	8.03E+02	F23	AVG	**2.60E+03**	2.60E+03	2.64E+03	2.61E+03
	SD	8.23E−01	**8.17E+02**	8.92E+00	1.58E+00		SD	**1.65E+00**	1.58E+00	4.20E+00	2.37E+00
F9	AVG	**9.00E+02**	**9.00E+02**	**9.00E+02**	**9.00E+02**	F24	AVG	**2.68E+03**	2.69E+03	2.74E+03	2.69E+03
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**		SD	**9.32E+01**	8.50E+01	1.16E+01	8.96E+01
F10	AVG	1.04E+02	**1.02E+03**	2.43E+03	1.12E+03	F25	AVG	2.92E+03	2.92E+03	2.94E+03	**2.90E+03**
	SD	5.53E+01	**3.17E+01**	1.85E+02	8.90E+01		SD	2.26E+01	2.29E+01	1.18E+01	**1.18E+01**
F11	AVG	**1.10E+03**	**1.10E+03**	1.36E+03	**1.10E+03**	F26	AVG	**2.90E+03**	**2.90E+03**	3.31E+02	**2.89E+03**
	SD	**0.00E+00**	**0.00E+00**	2.24E+02	**3.03−01**		SD	**0.00E+00**	**0.00E+00**	1.65E+01	**1.82E+01**
F12	AVG	1.29E+03	1.34E+03	5.52E+06	**1.20E+03**	F27	AVG	3.09E+03	3.09E+03	3.12E+03	**3.08E+03**
	SD	5.66E+01	7.43E+01	3.77E+06	**1.13−01**		SD	1.96E+00	7.96E−01	5.64E+01	**7.35−01**
F13	AVG	1.30E+03	**1.30E+03**	9.22E+04	1.30E+03	F28	AVG	3.14E+03	**3.11E+03**	3.28E+03	3.10E+03
	SD	2.54E+00	**2.50E+00**	9.74E+04	2.04E+00		SD	9.41E+01	**5.84E+01**	6.97E+00	0.00E+00
F14	AVG	**1.40E+03**	**1.40E+03**	7.67E+04	**1.40E+03**	F29	AVG	3.13E+03	3.1314E+03	**2.29E+03**	3.13E+03
	SD	**1.82−01**	**4.38−01**	3.39E+03	**5.65−01**		SD	1.90E+00	2.98E+00	**6.92E+01**	1.97E+00
F15	AVG	**1.50E+03**	**1.50E+03**	1.72E+04	1.50E+03	F30	AVG	5.73E+02	5.15E+04	3.43E+04	**3.40E+03**
	SD	**2.11−01**	**2.52−01**	1.70E+04	1.80E−02		SD	1.27E+04	1.94E+05	3.01E+04	**1.82E+01**
F16	AVG	**1.60E+03**	1.60E+03	1.72E+03	1.60E+03						
	SD	**3.32−01**	2.64E−01	5.11E+01	1.70E−01						

The bold values indicates the superior performance achieved by the corresponding algorithms

The convergence rate, measured in iterations, is another critical metric for assessing the efficiency of the algorithms. HSCAGOA achieves a convergence rate of 927 iterations for function F1, indicating a strong performance relative to other algorithms. For example, the SSA has a lower convergence rate of 821 iterations, suggesting that HSCAGOA may reach optimal solutions more swiftly. For function F2, HSCAGOA's convergence rate is 554 iterations, lower than DE's 924 iterations and SCA's 622 iterations. In function F7, HSCAGOA has a convergence rate of 337 iterations, significantly lower than DE's 964 iterations. For function F10, HSCAGOA achieves a convergence rate of 626 iterations, which is reasonable compared to DE's 950 iterations.

The best values achieved by each algorithm reflect their effectiveness in identifying optimal solutions. For instance, HSCAGOA secures a best value of 3.18267e+03 for function F1, substantially outperforming SCA, which achieves the best value of 2.069e+03. This trend of superior performance continues across the other functions, with HSCAGOA consistently providing better results compared to its counterparts, including BBO and AOA. Furthermore, the average values across multiple runs indicate the robustness of the solutions obtained. Both HSCAGOA and DE achieve average values equivalent to their best values for function F1, demonstrating their ability to consistently find optimal solutions.

For function, F2, HSCAGOA achieves a best value of 4.81634e+02, which is competitive, although DE achieves a slightly better best value of 4.0756e+02. The average value for HSCAGOA aligns with its best value, reinforcing its reliability. In function F7, HSCAGOA achieves a best value of 2.5057e+03, comparable to DE's best value of 2.501e+03, with the average value again matching the best value. Similarly, for function F10, HSCAGOA achieves a best value of 2.5057e+03, which is comparable to DE's best value of 2.501e+03, and the average value further confirms its reliability. The standard deviation (SD) is an important measure of the variability of results and is critical for assessing the reliability of each algorithm. A lower SD indicates that an algorithm produces stable results across different runs. In this analysis, all algorithms exhibit an SD of 0 for the best and average values, suggesting that they consistently achieve the same results across multiple trials for the respective functions. This consistency is particularly vital in optimization scenarios where reliability is important.

In conclusion, the comparison metrics presented in Table 17 offer valuable insights into the performance of various optimization algorithms on the CEC 2022 benchmark problems. HSCAGOA demonstrates superior efficiency in runtime and convergence rate while consistently achieving high-quality solutions.

5.8 Sensitivity Analysis

This section presents a sensitivity analysis of the HSCAGOA using the CEC 2022 benchmark test suite (see in Table 18). The primary objective of this analysis is to examine how variations in population size and maximum iteration counts affect the algorithm's performance. By systematically adjusting these parameters, we aim to gain insights into their influence on convergence speed and solution quality across different optimization scenarios.

In the sensitivity analysis, we explore a range of population sizes, specifically $N = 20, 30, 50, 100$, and three maximum iteration limits set at 1000, 2000, 3000. This structured approach allows us to evaluate the algorithm's responsiveness to changes in its configuration, thereby identifying optimal settings for effective performance. The results of the sensitivity analysis are summarized in Table 19. This table provides a comprehensive overview of the average performance (AVG) and standard deviation (SD) of HSCAGOA across various benchmark functions. Each function's performance is assessed under different combinations of population sizes and maximum iterations, offering a clear view of how these parameters interact.

1. For Function F1, the algorithm consistently achieves an average performance of $3.00E+02$ across all tested configurations, indicating that HSCAGOA maintains reliable performance regardless of population size or iteration limits. The low standard deviation suggests minimal variability in results.
2. In Function F2, a gradual decrease in average performance is observed as the population size increases. The standard deviations also vary, particularly at smaller sizes, indicating that larger populations may enhance stability in results.

 https://doi.org/10.1007/s44196-025-00823-6

Table 17 Performance comparison of optimization algorithms on CEC 2022 benchmark problems

Function	Algorithm	Runtime (s)	Convergence rate (iterations)	Best value	Average value	SD value
F1	HSCAGOA	1.283023	927	3.18267e+03	3.18267e+03	0
	SCA	1.31699	936	2.069e+03	2.069e+03	0
	DE	1.814008	983	3.7334e+03	3.7334e+03	0
	BBO	3.586927	971	300.0312	300.0312	0
	AOA	1.335857	575	7.4311e+03	7.4311e+03	0
	DMOA	2.798424	883	9.7812e+02	9.7812e+02	0
	GWO	1.34673	999	4.3747e+02	4.3747e+02	0
	SSA	1.248462	821	3.00e+02	3.00e+02	0
F2	HSCAGOA	0.670774	554	4.81634e+02	4.81634e+02	0
	SCA	0.706245	622	4.6073e+02	4.6073e+02	0
	DE	1.912726	924	4.0756e+02	4.0756e+02	0
	BBO	50.19411	960	404.0455	404.0455	0
	AOA	0.785240	625	1.2646e+03	1.2646e+03	0
	DMOA	3.834897	988	4.054e+02	4.054e+02	0
	GWO	0.838576	998	4.1234e+02	4.1234e+02	0
	SSA	1.114214	970	4.0608e+02	4.0608e+02	0
F7	HSCAGOA	0.681301	337	2.0628e+03	2.0628e+03	0
	SCA	1.108419	903	2.0664e+03	2.0664e+03	0
	DE	0.963404	964	2.0034e+03	2.0034e+03	0
	BBO	2.49987	998	2.0045e+03	2.0045e+03	0
	AOA	0.90079	732	2.0694e+03	2.0694e+03	0
	DMOA	2.750756	989	2.0233e+03	2.0233e+03	0
	GWO	1.221765	1000	2.0254e+03	2.0254e+03	0
	SSA	0.9518785	963	2.0369e+03	2.0369e+03	0
F10	HSCAGOA	1.16878	626	2.5057e+03	2.5057e+03	0
	SCA	0.994458	721	2.6065e+03	2.6065e+03	0
	DE	1.509375	950	2.501e+03	2.501e+03	0
	BBO	2.984024	723	2.5004e+03	2.5004e+03	0
	AOA	1.444756	509	2.7028e+03	2.7028e+03	0
	DMOA	3.064984	550	2.5009e+03	2.5009e+03	0
	GWO	1.176807	953	2.5003e+03	2.5003e+03	0
	SSA	69.610738	924	2.5006e+03	2.5006e+03	0

3. Function F3 exhibits a uniform average performance of $6.00E + 02$ across all configurations, demonstrating that HSCAGOA can reliably solve this function with minimal fluctuation in results.
4. For Function F4, average values remain relatively stable, but the standard deviations indicate some variability, particularly at lower population sizes. This suggests that while the algorithm performs well, performance can still be influenced by the choice of parameters.
5. Functions F5 and F6 show strong consistency in their average performance, with F5 averaging $9.00E + 02$ and F6 at $1.80E + 03$. The low standard deviations reinforce the reliability of the algorithm's performance across different configurations.
6. In Function F7, while the average results are consistent, the standard deviations increase with larger population sizes, indicating that the algorithm's performance may exhibit more variability under these conditions.

Table 18 Description of CEC 2022 test suite

Function type	No.	Name	[lb, ub]	Fmin
Unimodal function	F1	Shifted and full rotated Zakharov function	[−100, 100]	300
Basic functions	F2	Shifted and full rotated Rosenbrock's function	[−100, 100]	400
	F3	Shifted and full rotated expanded Schaffer's F6 function	[−100, 100]	600
	F4	Shifted and full rotated non-continuous Rastrigin's function	[−100, 100]	800
	F5	Shifted and rotated Levy function	[−100, 100]	900
Hybrid functions	F6	Hybrid function 1 ($N = 3$)	[−100, 100]	1800
	F7	Hybrid function 2 ($N = 6$)	[−100, 100]	2000
	F8	Hybrid function 3 ($N = 5$)	[−100, 100]	2200
Composite functions	F9	Composite function 1 (N = 5)	[−100, 100]	2300
	F10	Composite function 2 ($N = 4$)	[−100, 100]	2400
	F11	Composite function 3 ($N = 5$)	[−100, 100]	2600
	F12	Composite function 4 ($N = 6$)	[−100, 100]	2700

7. Functions F8 through F12 demonstrate stable average performance, with slight fluctuations in standard deviations, particularly when larger population sizes are utilized. This stability suggests that HSCAGOA can effectively navigate the solution space across a variety of functions.

In conclusion, the sensitivity analysis highlights the robustness of HSCAGOA under different configurations. The algorithm maintains strong performance across various parameter settings, with certain functions exhibiting remarkable stability.

5.9 Applications of HSCAGOA in Engineering Design Problem Solving

This subsection evaluates the effectiveness of the HSCAGOA algorithm in addressing engineering design problems, specifically constrained optimization challenges. A total of six engineering problems have been used for this assessment: the welded beam design problem, the pressure vessel design problem, the cantilever beam design problem, the tension/compression spring design problem, the gear train design problem, and the three-bar truss design problem. Due to the numerous inequality constraints in these problems, if any of these constraints are violated, algorithms typically employ a penalty function to achieve a feasible solution. The results obtained by HSCAGOA are compared with those produced by other algorithms to demonstrate its efficacy.

5.9.1 Welded Beam Design Problem

The goal of the welded beam design problem is to minimise the weight of a welded beam while conforming to four sets of constraints: beam height (t), connecting beam thickness (b), weld width (h), and connecting beam length (l). Figure 7 provides a detailed illustration of each variable. The following variables affect the constraints: beam bending load (P), beam deflection (δ), shear stress (τ), and bending stress (σ).pg

Table 19 Sensitivity analysis of HSCAGOA on the CEC 2022 test suite

Functions	Values	Max iterations = 1000				Max iterations = 2000				Max iterations = 3000			
		N = 20	N = 30	N = 50	N = 100	N = 20	N = 30	N = 50	N = 100	N = 20	N = 30	N = 50	N = 100
F1	AVG	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02
	SD	3.12E−03	2.52E−07	1.66E−05	1.91E−04	6.62E−05	0.00E+00	7.54E−14	1.62E−11	2.94E−04	0.00E+00	0.00E+00	0.00E+00
F2	AVG	4.05E+02	4.04E+02	4.02E+02	4.01E+02	4.05E+02	4.03E+02	4.02E+02	4.01E+02	4.05E+02	4.04E+02	4.02E+02	4.01E+02
	SD	2.09E+00	1.55E+00	2.01E+00	1.49E+00	2.47E+00	1.82E+00	1.99E+00	1.51E+00	2.41E+00	2.07E+00	1.99E+00	1.51E+00
F3	AVG	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02
	SD	1.09E−01	6.85E−02	7.02E−02	1.58E−01	2.50E−02	1.29E−02	3.49E−03	1.89E−02	6.80E−03	4.83E−03	2.09E−04	2.91E−03
F4	AVG	8.08E+02	8.08E+02	8.06E+02	8.06E+02	8.06E+02	8.06E+02	8.05E+02	8.04E+02	8.08E+02	8.06E+02	8.04E+02	8.03E+02
	SD	2.74E+00	3.19E+00	2.37E+00	1.48E+00	2.00E+00	2.64E+00	1.75E+00	1.46E+00	3.13E+00	3.13E+00	1.43E+00	1.28E+00
F5	AVG	9.01E+02	9.00E+02	9.00E+02	9.00E+02	9.01E+02	9.00E+02	9.00E+02	9.00E+02	9.01E+02	9.00E+02	9.00E+02	9.00E+02
	SD	7.31E−01	4.12E−03	1.02E−02	3.81E−02	1.10E+00	8.29E−02	1.11E−09	5.05E−05	1.27E+00	1.00E−01	0.00E+00	3.86E−11
F6	AVG	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03	1.80E+03
	SD	4.99E−01	1.59E−01	2.90E−01	3.61E−01	6.10E−01	1.08E−01	8.51E−02	7.44E−02	4.87E−01	1.23E−01	7.63E−02	4.61E−02
F7	AVG	2.01E+03	2.01E+03	2.01E+03	2.01E+03	2.01E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03
	SD	8.27E+00	7.80E+00	6.01E+00	7.27E+00	8.55E+00	5.05E+00	3.59E+00	1.74E+00	6.76E+00	6.91E−01	3.59E+00	5.89E−01
F8	AVG	2.21E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03
	SD	7.60E+00	1.57E+00	1.26E+00	8.37E−01	4.91E+00	6.66E−01	5.46E−01	3.55E−01	6.45E+00	5.10E−01	4.65E−01	3.71E−01
F9	AVG	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03
	SD	1.20E−07	9.50E−09	6.65E−08	1.32E−07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	AVG	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03
	SD	6.58E−02	4.44E−02	5.58E−02	4.00E−02	1.94E+01	5.70E−02	3.80E−02	4.46E−02	2.05E+01	4.23E−02	4.21E−02	3.23E−02
F11	AVG	2.66E+03	2.60E+03	2.60E+03	2.60E+03	2.64E+03	2.60E+03	2.60E+03	2.60E+03	2.62E+03	2.61E+03	2.60E+03	2.60E+03
	SD	9.99E+01	1.54E−02	6.61E−02	1.91E−01	6.46E+01	9.94E−08	3.99E−06	9.41E−05	6.52E+01	5.48E+01	2.57E−10	2.78E−08
F12	AVG	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03
	SD	1.49E+00	1.50E+00	3.70E−01	3.40E−02	1.82E+00	1.36E+00	5.80E−01	6.74E−04	1.86E+00	1.42E+00	3.78E−01	4.09E−07

The objective function $f(z)$ aims to minimize the weight of the beam and is defined as:

$$
\begin{aligned}
&[h, l, t, b] = [z_1, z_2, z_3, z_4] \\
&f(z) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14.0 + z_2)
\end{aligned}
\tag{17}
$$

where $z = [z_1, z_2, z_3, z_4]$.

The design problem includes the following constraints:

$$c_1(\vec{z}) = -(\tau_{\max} - \tau(\vec{z})) \leq 0 \tag{18}$$

$$c_2(\vec{z}) = \sigma(\vec{z}) - \sigma_{\max} \leq 0 \tag{19}$$

$$c_3(\vec{z}) = \delta(\vec{z}) - \delta_{\max} \leq 0 \tag{20}$$

$$c_4(\vec{z}) = -(z_4 - z_1) \leq 0 \tag{21}$$

$$c_5(\vec{z}) = P - P_c(\vec{z}) \leq 0 \tag{22}$$

$$c_6(\vec{z}) = 0.125 - z_1 \leq 0 \tag{23}$$

$$c_7(\vec{z}) = 1.10471 z_1 + 0.04811 z_3 z_4 (14.0 + z_2) - 5.0 \leq 0 \tag{24}$$

The shear stress (τ) is given by:

$$\tau(\vec{z}) = \sqrt{\tau'^2 + 2\tau'\tau'' \frac{z_2}{2R} + \tau''^2} \tag{25}$$

where

$$\tau' = \frac{P}{\sqrt{2} z_1 z_2}, \quad \tau'' = \frac{MR}{J} \tag{26}$$

The bending stress (σ) is given by:

$$\sigma(\vec{z}) = \frac{6PL}{z_4 z_2^3} \tag{27}$$

The beam deflection (δ) is given by:

$$\delta(\vec{z}) = \frac{6PL^3}{E z_4 z_2^3} \tag{28}$$

The critical load (P_c) is given by:

$$P_c(\vec{z}) = \frac{4.013 E z_3^2 z_4^6}{L^2} \left(1 - \frac{z_3}{2L} \sqrt{\frac{E}{4G}}\right) \tag{29}$$

- $P = 6000$ lb (bending load)
- $L = 14$ in (connecting beam length)
- $E = 30 \times 10^6$ psi (modulus of elasticity)
- $\tau_{\max} = 13600$ psi (maximum shear stress)
- $\sigma_{\max} = 30000$ psi (maximum bending stress)
- $\delta_{\max} = 0.25$ in (maximum beam deflection)
- $0.1 \leq z_i \leq 2$ for $i = 1, 4$
- $0.1 \leq z_i \leq 10$ for $i = 2, 3$

By optimizing these variables under the given constraints, the goal is to achieve a lightweight and structurally sound welded beam design.

Fig. 7 Structure of Welded beam

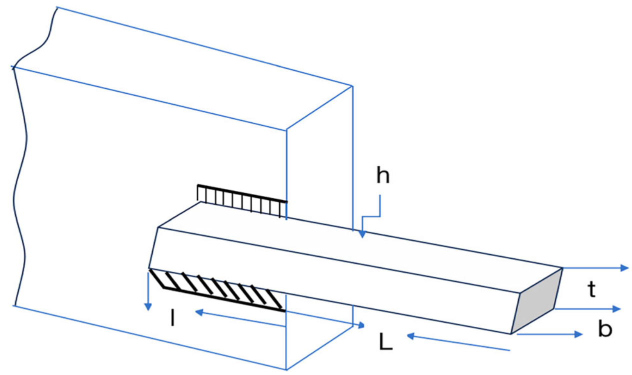


Table 20 Comparative results of optimization algorithms for Welded beam design

Algorithms	h	l	t	b	Best	Worst	Average	SD
HSCAGOA	0.2151	3.8235	8.9652	0.2088	1.8771	1.9645	1.8869	0.0329
SCA	0.1803	4.3572	8.9492	0.2180	1.8797	1.9873	1.8500	0.0363
DE	0.7852	2	2	2	1.089E+14	1.089E+14	1.089E+14	0.0477
CPSOGSA	0.1399	4.982	9.055	0.2056	1.7194	2.6981	1.9239	0.2171
BBO	1.148	1.298	2	2	1.089E+14	1.089E+14	1.089E+14	0.1372
AOA	0.1407	9.4513	10	0.2025	2.4909	2.4909	2.4909	0

The penalized objective function, represented as Φ, is formulated as follows:

$$\Phi = f(z) + \text{PCONST} \times \sum_i \max(0, c_i)^2 \tag{30}$$

In Eq. (30), the term $f(z)$ denotes the original objective function's value, while PCONST set at (10^5) serves as the constant for the penalty function. The expressions $\max(0, c_i)$ for $(i = 1, 2, 3, \ldots)$ represent any constraints that have been violated, with only positive violations contributing to the penalty. This approach guarantees that any violations of the constraints are squared and multiplied by the penalty constant, effectively steering the optimization process toward feasible solutions.

Table 20 presents the comparative results for the Welded Beam Design problem solved using various optimisation algorithms. Among them, HSCAGOA achieves notable performance with respect to the objective function and design variables. For instance, HSCAGOA obtains a best function value of 1.8771, which is competitive compared to other algorithms such as SCA and CPSOGSA. Moreover, HSCAGOA demonstrates consistent performance, as indicated by its average value of 1.8869 and standard deviation of 0.0329, suggesting robustness in finding solutions across multiple runs. This table underscores HSCAGOA's efficacy in optimising the welded beam design problem.

5.9.2 Pressure Vessel Problem

A pressure vessel, essential for containing gases or liquids under high pressure, is typically structured with cylindrical ends capped by hemispherical heads (Fig. 8). This optimization challenge, originally posed by Sandgren [47], adheres to the ASME boiler and pressure vessel code, ensuring it meets stringent engineering standards.

Mathematically, the problem is formulated as follows [24]:

$$Let\ \vec{z} = [T_s, T_h, R, L] \tag{31}$$

$$\text{Minimize } f(\vec{z}) = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_h^2 L \tag{32}$$

subject to

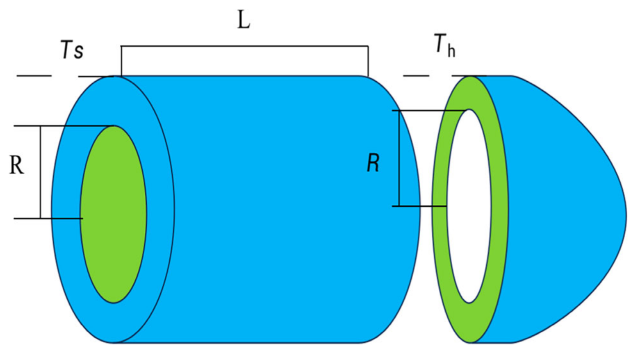


Fig. 8 Structure of pressure vessel

Table 21 Comparative results of optimization algorithms for pressure vessel design

Algorithms	T_s	T_h	R	L	Best	Worst	Average	SD
HSCAGOA	0	0	40.344	200	4.9143e+03	7.5640E+03	6.9350E+03	937.2211
SCA	0.439	0	40.674	196.384	5.1237E+03	7.5640E+03	6.9832E+03	1.0258E+03
DE	10	10	53.6191	71.997	2.0432E+05	2.0432E+05	2.0432E+05	0.1992
CPSOGSA	0	0.2401	40.32	200	4.5273E+03	8.1131E+03	5.0156E+03	684.4933
BBO	10	10	55.98	57.01	2.0432E+05	2.1112E+05	2.0583E+05	1.7601E+03
AOA	0	0	40.35	200	7.2952E+03	7.2952E+03	7.2952E+03	0

$$g_1(\vec{z}) = -T_s + 0.0193R \leq 0, \tag{33}$$
$$g_2(\vec{z}) = -T_h + 0.00954R \leq 0, \tag{34}$$
$$g_3(\vec{z}) = -\pi R^2 L - \frac{4}{3}\pi R^3 + 750 \times 11728 \leq 0, \tag{35}$$
$$g_4(\vec{z}) = L - 240 \leq 0, \tag{36}$$
$$0 \leq T_s, T_h \leq 99, 0 \leq R, L \leq 200 \tag{37}$$

The penalty function, represented as Φ, is formulated as follows:

$$\Phi = f(\vec{z}) + \text{PCONST} \times \sum_i \max(0, g_i)^2 \tag{38}$$

In Eq. (38), $f(\vec{z})$ denotes the value of the original objective function, while PCONST is the penalty constant, assigned a value of 10^4. The expression $\max(0, g_i)$ (for $i = 1, 2, 3, \ldots$) captures any constraints that have been violated, ensuring that only those violations that exceed zero are factored into the penalty. This approach effectively directs the optimization process toward feasible solutions by imposing penalties for any constraint breaches.

Table 21 presents comparative results for solving the Pressure Vessel Design Optimisation Problem using various algorithms. HSCAGOA achieves a notable best objective function value of 4914.3, with corresponding optimal design variables ($T_s = 0$, $T_h = 0$, $R = 40.344$, $L = 200$). This algorithm outperforms others such as SCA, CPSOGSA, and AOA in terms of both solution quality and consistency. The worst-case scenario for HSCAGOA is 7564.0, indicating a relatively narrow range between best and worst outcomes, which is favourable for practical applications. The average solution across runs is 6935.0, with a standard deviation of 937.2211, reflecting HSCAGOA's robustness in maintaining stability across multiple optimizations. These results underscore HSCAGOA's efficacy in minimising costs associated with welding, material usage, and forming processes in pressure vessel design.

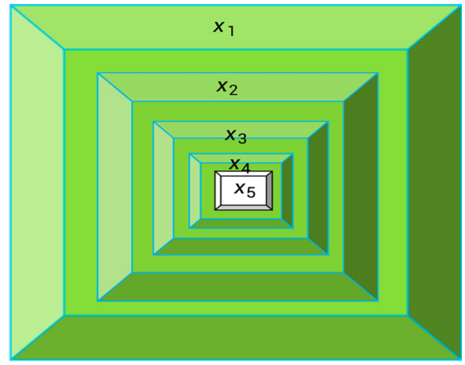


Fig. 9 Structure of Cantilever beam

Table 22 Comparison of algorithm performance for Cantilever beam design

Algorithms	x_1	x_2	x_3	x_4	x_5	Best	Worst	AVG	SD
HSCAGOA	5.686	5.376	3.844	2.732	2.4954	1.3097	1.4198	1.3473	0.0231
SCA	5.886	5.4115	3.88	3.308	1.801	1.3083	1.3413	1.3244	0.0092
DE	5.6943	5.0253	4.251	3.314	2.038	1.3004	1.3004	1.3004	2.2584E−16
CPSOGSA	5.6943	5.0253	4.251	3.314	2.0375	1.3004	1.3004	1.3004	1.3039E−16
BBO	5.6943	5.0253	4.251	3.314	2.038	1.3004	1.3004	1.3004	2.4032E−12
AOA	5.9063	5.2932	5.8172	2.4024	2.7738	1.4224	1.4224	1.4224	2.4032E−12

5.9.3 Cantilever Beam Design Problem

The cantilever beam design problem involves a beam with a vertical load applied at its free end while the other end is rigidly supported. Figure 9 depicts the structure of this problem. The objective is to minimize the weight of the beam, ensuring that the vertical displacement does not exceed the specified limit for the optimal design.

The mathematical formulation of the cantilever beam design problem is as follows:

$$\text{Minimize } f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \tag{39}$$

subject to:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{40}$$

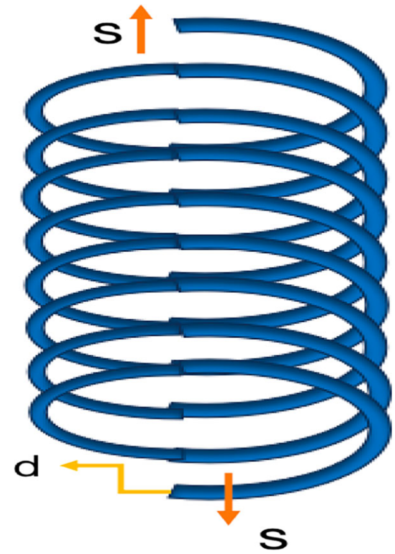
$$0.01 \leq x_i \leq 100;\; i = 1, 2, \ldots, 5 \tag{41}$$

Table 22 presents a comparative analysis of various optimisation algorithms applied to the Cantilever Beam Design problem. Among these algorithms, HSCAGOA achieves competitive performance with notable results. It minimises the objective function effectively with optimal values of x_1, x_2, x_3, x_4, and x_5: 5.686, 5.376, 3.844, 2.732, and 2.4954, respectively. The algorithm exhibits the best solution of 1.3097, a worst solution of 1.4198, an average solution of 1.3473, and a standard deviation of 0.0231. These metrics demonstrate HSCAGOA's robustness in finding solutions close to optimal across multiple runs, indicating its effectiveness in addressing the complex design constraints and objectives of the Cantilever Beam problem compared to other algorithms listed.

5.9.4 Tension/Compression Spring Design Problem

Optimising the design of tension/compression springs is a complex problem in engineering, involving lowering the weight of the springs while following four particular restrictions relating to buckling, stress, and deflection [24]. The schematic representation of this design problem is shown in Fig. 10. The objective is to find the optimal values for the cross-sectional areas of the truss bars.

Fig. 10 Design of tension/compression spring



The optimization problem for designing tension/compression springs can be mathematically formulated as follows:

$$\text{Let } [z] = [z_1, z_2, z_3] = [d, D, P] \tag{42}$$

$$\text{Minimize } f(z) = (z_3 + 2)z_2 z_1^2 \tag{43}$$

subject to:

$$g_1(z) = 1 - \frac{z_2^3 z_3}{71785 z_1^4} \leq 0 \tag{44}$$

$$g_2(z) = \frac{4z_2^2 - z_1 z_2}{12566(z_2 z_1^3)} + \frac{1}{5108 z_1^2} - 1 \leq 0 \tag{45}$$

$$g_3(z) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \leq 0 \tag{46}$$

$$g_4(z) = \frac{z_1 + z_2}{1.5} - 1 \leq 0 \tag{47}$$

$$0.05 \leq z_1 \leq 2.00;\ 0.25 \leq z_2 \leq 1.30;\ 2.00 \leq z_3 \leq 15.00 \tag{48}$$

Table 23 provides a comprehensive analysis of various optimisation algorithms' performance in solving the tension/compression spring design problem. Focusing on the HSCAGOA algorithm, it demonstrates superior performance in optimising the design variables (d, D, P) to achieve a minimum weight for the spring. Specifically, HSCAGOA achieves a best fitness value of 3.6678 and a worst fitness value of 3.9053, with an average (AVG) fitness value of 3.7368 and a standard deviation (SD) of 0.0571. These results indicate not only the efficacy of HSCAGOA in finding optimal solutions but also its consistency, as evidenced by the relatively low standard deviation. Compared to other algorithms like SCA, DE, CPSOGSA, BBO, and AOA, HSCAGOA provides a balanced and effective optimisation, making it a promising choice for solving this design problem.

5.9.5 Gear Train Design Problem

The primary goal is to reduce the cost of the gear ratio in a gear train by optimising four design variables, which correspond to the number of teeth on the gears.

$$\text{Gear ratio} = \frac{t_A \times t_B}{t_C \times t_D} \tag{49}$$

 https://doi.org/10.1007/s44196-025-00823-6

Table 23 Performance comparison of algorithms for tension/compression spring design

Algorithms	d	D	P	Best	Worst	AVG	SD
HSCAGOA	0.1357	1.2577	12.1516	3.6678	3.9053	3.7368	0.0571
SCA	0.1388	1.3	11.9236	3.6626	3.7373	3.6959	0.0227
DE	2	2	2	409.775	410.5944	409.8023	0.1496
CPSOGSA	0.1391	1.300	11.8924	3.6619	3.7298	3.6656	0.0132
BBO	2	2	2	409.775	409.8971	409.8099	0.0385
AOA	0.1432	1.2407	14.5844	4.5767	4.5767	4.5767	0

Fig. 11 Structure of a gear train design

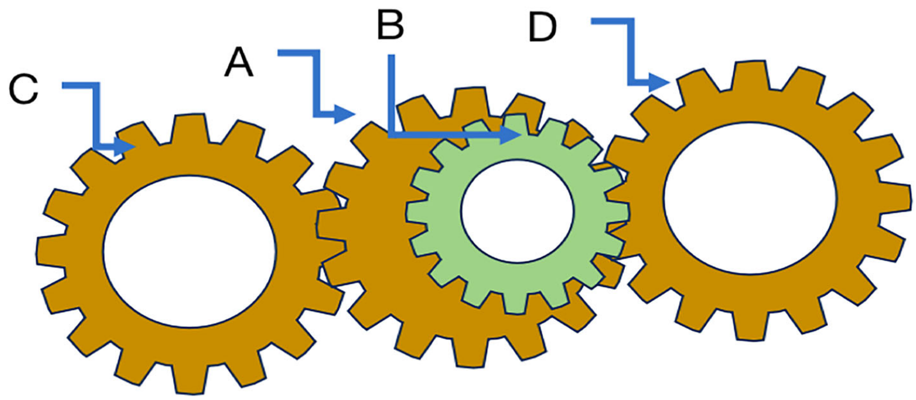


Table 24 Comparing the results for the gear train design issue

Algorithms	n_A	n_B	n_C	n_D	Best	Worst	AVG	SD
HSCAGOA	51.5940	14.7012	12	23.5816	2.70E−12	1.8274E−08	1.805E−09	3.2364E−09
SCA	58.1032	12	29.7673	42.7937	2.7009E−12	8.7008E−09	1.4795E−09	1.6764E−09
DE	43.17577	30.5385	12	60	2.70E−12	1.3616E−09	6.2675E−10	5.2289E−10
CPSOGSA	51.3177	28.867	12.2773	43.5493	2.70E−12	4.4744E−08	1.0894E−08	1.1842E−08
BBO	52.8697	16.3518	21.6099	46.0879	2.70E−12	2.4573E−07	2.0974E−08	4.5887E−08
AOA	22.5614	12	13.0271	46.5477	9.9216E−10	9.9216E−10	9.9216E−10	9.90E−03

The gear ratios t_A, t_B, t_C, and t_D determine the performance of the gear train, as depicted in Fig. 11. Each variable ranges from 12 to 60 and is constrained to integer values. The mathematical formulation of the gear train design problem [47] is given by:

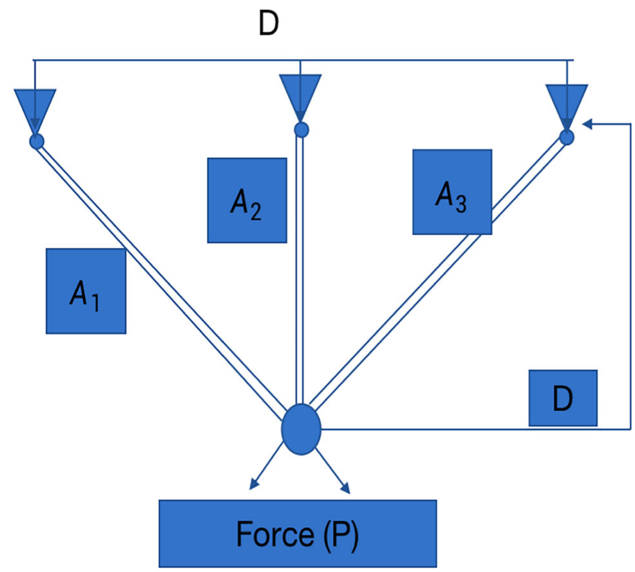
$$\begin{aligned} &Let\ [t_A, t_B, t_C, t_D] \\ &\text{Minimize} \quad f(t_A, t_B, t_C, t_D) = \left(\frac{1}{6.931} - \frac{t_A \times t_B}{t_C \times t_D}\right)^2 \end{aligned} \tag{50}$$

subject to:

$$12 \leq t_A, t_B, t_C, t_D \leq 60 \tag{51}$$

Table 24 provides comparative results of various optimisation algorithms applied to this problem. HSCAGOA achieves superior performance, with optimised values of $t_A = 51.5940$, $t_B = 14.7012$, $t_C = 12$, and $t_D = 23.5816$. It demonstrates the best solution of 2.70×10^{-12}, the worst solution of 1.8274×10^{-08}, the average solution of 1.805×10^{-09}, and the standard deviation of 3.2364×10^{-09}. These metrics underscore the effectiveness of HSCAGOA in finding near-optimal solutions consistently compared to the other algorithms listed.

Fig. 12 Design of three-bar truss



5.9.6 Three-Bar Truss Design Problem

As seen in Fig. 12, a three-bar planar truss structure is taken into consideration in this design. The purpose is to reduce the relevant weights to a minimum. This scenario has two optimisation variables ($A_1(=x_1)$ and $A_2(=x_2)$ with three constraints: stress, deflection, and buckling.

The objective function $f(A_1, A_2)$ aims to minimize the weight and is defined as:

$$\min f(A_1, A_2) = l\left(2\sqrt{2}x_1 + x_2\right) \tag{52}$$

The design problem includes the following constraints:

$$G_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \tag{53}$$

$$G_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \tag{54}$$

$$G_3 = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0 \tag{55}$$

- $l = 100$ cm (length of the truss)
- $P = 2\frac{\mathrm{kN}}{\mathrm{cm}^2}$ (load)
- $\sigma = 2\frac{\mathrm{kN}}{\mathrm{cm}^2}$ (stress)
- $0 \leq x_1, x_2 \leq 1$ (range for optimization variables)

Table 25 presents the results for the three-bar truss design problem across various optimisation algorithms, focusing on the performance of HSCAGOA. This problem aims to optimise the design variables x_1 and x_2 to minimise the objective function related to the truss structure. HSCAGOA achieves a notable best solution of 106.934 with a very consistent performance, as indicated by its narrow range between the best (106.934) and worst (106.955) solutions. The average solution of 106.938 and standard deviation of 0.0043 further highlight its stability and accuracy in finding optimal or near-optimal solutions compared to other algorithms listed. This suggests that HSCAGOA is effective in handling the specific complexities and constraints of the three-bar truss design problem, offering reliable performance across multiple runs.

Table 25 Results estimated for three-bar truss design problem

Algorithm	x_1	x_2	Best	Worst	Average	SD
HSCAGOA	0.2187	0.1885	106.934	106.955	106.938	0.0043
SCA	0.22	0.1885	106.9344	106.9399	106.9357	0.0014
DE	0.2195	0.1884	106.9344	106.9344	106.9344	3.0088E−14
CPSOGSA	0.2645	0.3651	107.0176	147.5218	116.3557	8.7654
BBO	0.2195	0.1884	106.9344	106.9344	106.9344	3.8423E−13
AOA	0.2197	0.1882	106.9345	106.9345	106.9345	0

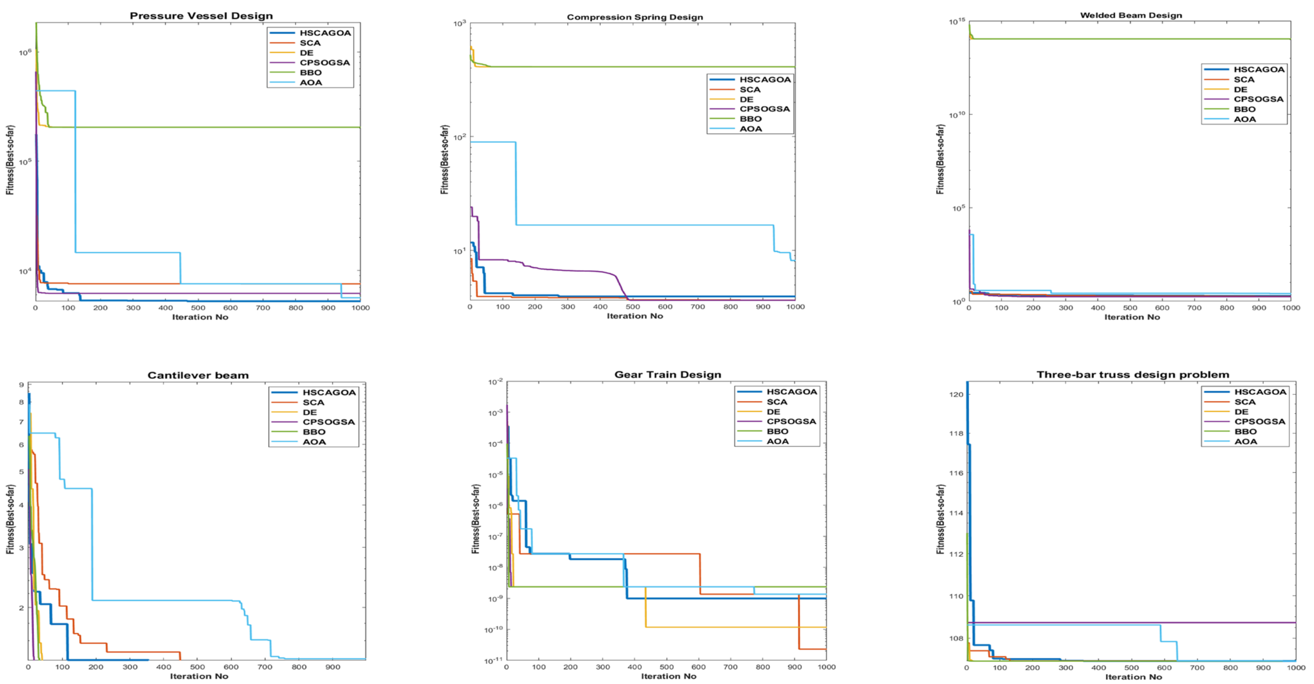


Fig. 13 Convergence comparison of HSCAGOA and competitor algorithms for engineering design problems

5.9.7 Assessment of Convergence Curves in Engineering Design Applications

The performance comparison of the HSCAGOA against several other algorithms across six distinct engineering design problems reveals significant insights. Graphically (Fig. 13), the y-axis depicts fitness (best-so-far), with lower scores indicating superior performance, while the x-axis represents iterations. Across all six design problems, HSCAGOA consistently exhibits faster convergence to solutions compared to most other algorithms. Specifically:

- In the pressure vessel design problem, HSCAGOA outperforms all other algorithms by converging to a notably superior solution.
- For the tension/compression spring design problem, HSCAGOA achieves a better solution than all algorithms except CPSOGSA.
- In the welded beam design problem, HSCAGOA's convergence is superior to all other algorithms.
- Similarly, in the cantilever beam design problem, HSCAGOA converges to a better solution than all other algorithms.
- In the gear train design problem, HSCAGOA demonstrates superior convergence, although it slightly trails behind SCA and DE.
- For the three-bar truss design problem, HSCAGOA again proves its effectiveness by converging to a better solution than all other algorithms.

The convergence patterns observed in the graphs underscore HSCAGOA's competitiveness and effectiveness across diverse engineering design challenges. These findings highlight HSCAGOA as a robust choice for optimizing complex engineering problems, showcasing its ability to consistently approach optimal solutions efficiently compared to its peers.

5.10 Impact of Population Diversity on Convergence Behavior of HSCAGOA

In optimization problems, particularly in engineering design applications, the balance between exploration and exploitation is crucial for achieving optimal solutions. The HSCAGOA algorithm demonstrates a unique ability to maintain population diversity, significantly influencing its convergence behaviour compared to other algorithms. This section discusses how diversity impacts the performance of HSCAGOA across various case studies.

5.10.1 Diversity in HSCAGOA

HSCAGOA employs mechanisms that promote diversity within the population of candidate solutions. By integrating the Gazelle Optimization Algorithm's exploitation strategy and incorporating Brownian motion and Lévy flight mechanisms, HSCAGOA effectively explores the solution space while refining promising solutions. This approach allows the algorithm to avoid premature convergence.

5.10.2 Case Studies Illustrating Diversity's Impact

The following case studies highlight the relationship between population diversity and convergence behaviour in HSCAGOA:

- Welded Beam Design Problem: In this case, HSCAGOA consistently achieved optimal solutions with minimal standard deviation across multiple runs. The diversity maintained within the population allowed the algorithm to explore various potential designs effectively, leading to a robust solution that minimized the beam's weight while adhering to structural constraints. In contrast, other algorithms, such as SCA, exhibited higher variability in their results, indicating a tendency towards local optima.
- Pressure Vessel Design Problem: HSCAGOA's ability to maintain diversity resulted in a best objective function value of 4914.3, significantly outperforming other algorithms. The diverse population facilitated the exploration of different design configurations, which enabled HSCAGOA to navigate the complex constraint landscape more effectively. Other algorithms, such as DE, struggled to maintain diversity, resulting in suboptimal solutions.
- Cantilever Beam Design Problem: The convergence behaviour of HSCAGOA was notably superior, with a best solution of 1.3097 and a low standard deviation. The diversity within the population allowed HSCAGOA to explore various design variables effectively, leading to consistent performance across multiple iterations. In contrast, algorithms like CPSOGSA showed signs of premature convergence, resulting in less optimal solutions.
- Three-Bar Truss Design Problem: HSCAGOA achieved a best solution of 106.934 with a narrow range between best and worst outcomes, indicating stability. The ability to maintain diversity in the population was instrumental in exploring the solution space thoroughly, allowing HSCAGOA to avoid local optima that other algorithms encountered.

The analysis of HSCAGOA's performance across various engineering design problems underscores the importance of population diversity in optimization algorithms. HSCAGOA's innovative mechanisms for maintaining diversity enhance its exploration capabilities and contribute to its superior convergence behaviour. By effectively balancing exploration and exploitation, HSCAGOA consistently approaches optimal solutions, demonstrating its robustness as a tool for solving complex engineering design challenges.

6 Analysis and Discussion of Experimental Results

The HSCAGOA consistently outperformed a spectrum of competing algorithms across both benchmark functions (CEC-2017) and practical engineering design problems. It exhibited notable strengths in terms of faster convergence rates and lower error rates, underscoring its efficiency in achieving optimal solutions. Across the extensive range of benchmark functions from F1 to F33, HSCAGOA demonstrated robust performance with rapid convergence and consistently superior solution quality compared to several established algorithms, including the classical SCA, DE, GSA, CPSOGSA, BBO, AOA, DMOA, PDO, GWO, and SSA. A significant advantage of HSCAGOA was its ability to achieve lower fitness values quicker than its competitors, often demonstrating notably lower error rates early in the optimisation process. This early convergence can be attributed to its hybrid nature, effectively combining the strengths of the SCA and the GOA. Importantly, regardless of the complexity of benchmark functions, HSCAGOA consistently maintained competitive performance, highlighting its robustness in tackling diverse optimisation landscapes.

In practical engineering applications such as welded beam design, pressure vessel design, cantilever beam design, tension/compression spring design, gear train design, and three-bar truss design, HSCAGOA exhibited versatility and effectiveness in optimising complex design parameters. It consistently delivered competitive results, effectively balancing performance metrics like stress minimization and structural stability. The adaptability of HSCAGOA was evident across both synthetic benchmark functions and real-world engineering problems, showcasing its potential utility in optimising complex systems with multiple constraints. Analysis of convergence curves revealed distinct patterns among algorithms, with HSCAGOA characterised by steep initial improvements and stable progress towards optimal solutions. This contrasted with the slower and less stable convergence observed in other algorithms, further highlighting the efficacy of HSCAGOA in efficiently navigating optimisation challenges. HSCAGOA emerges as a compelling hybrid optimisation approach, offering robust performance across various domains and demonstrating the significant potential for advancing optimisation capabilities in both theoretical benchmark settings and practical engineering applications.

Despite these successes, it is essential to acknowledge the limitations inherent in existing metaheuristic approaches. Many algorithms exhibit parameter sensitivity, where their performance is reliant on the selection of control parameters such as population size and mutation rates. This can complicate optimization, as finding the optimal settings requires extensive trial and error. Furthermore, achieving a proper balance between exploration and exploitation remains a challenge. Excessive focus on exploration can slow convergence, while too much exploitation may result in premature convergence. Finally, scalability issues frequently arise when tackling high-dimensional problems, as the effectiveness of algorithms diminishes with increasing problem complexity. Many traditional metaheuristic algorithms also struggle to effectively manage constraints present in real-world scenarios, limiting their applicability in practical optimization tasks. Addressing these limitations is crucial for improving the effectiveness and versatility of metaheuristic algorithms in complex optimization challenges.

6.1 Summary of the Obtained Results

The performance evaluation of the HSCAGOA across various benchmark functions and engineering design problems has yielded promising outcomes, demonstrating its robustness and effectiveness in optimization tasks. In the context of benchmark functions, HSCAGOA consistently exhibited superior convergence rates and solution quality compared to several established algorithms, including the classical Sine Cosine Algorithm (SCA), Differential Evolution (DE), and others. The analysis of convergence curves revealed that HSCAGOA reaches optimal solutions more quickly, often achieving lower error rates early in the optimization process. This efficiency is attributed to the hybrid nature of the algorithm, which effectively combines the exploration capabilities of the Sine Cosine Algorithm with the exploitation strategies of the Gazelle Optimization Algorithm. When applied to practical engineering design problems, HSCAGOA demonstrated its versatility to handle complex constraints. For instance, in the welded beam design problem, HSCAGOA achieved a competitive best function value while maintaining consistency across multiple runs, as indicated by a low standard deviation. In the pressure vessel design problem, it outperformed other algorithms, achieving a notable best objective function value of 4914.3. In the cantilever beam design problem, HSCAGOA again showed its effectiveness by converging to optimal design values. The results from the tension/compression spring design problem further confirmed HSCAGOA's reliability, with the algorithm producing a best fitness value of 3.6678 and demonstrating consistent performance metrics. In the gear train design problem, HSCAGOA maintained its competitive edge, and in the three-bar truss design problem, it exhibited strong convergence characteristics, yielding optimal solutions with minimal variability. The comprehensive evaluation across synthetic benchmarks and real-world engineering applications underscores HSCAGOA's potential as a powerful optimization tool. Its ability to deliver high-quality solutions while navigating complex design constraints positions it as a valuable resource in various optimization scenarios, paving the way for future applications in complex engineering problems.

7 Conclusion and Future Work

In conclusion, HSCAGOA represents a significant advancement in the field of metaheuristic optimization. By integrating the GOA's exploitation strategy with the robust global search capabilities of the SCA, HSCAGOA effectively addresses several inherent limitations of traditional optimisation methods. The innovative position-update mechanism of HSCAGOA, incorporating both sine and cosine functions along with adaptive behaviours inspired by gazelles, enhances its ability to navigate complex solution spaces. This dual-phase approach, emphasising exploration and exploitation during different stages of the optimisation process, ensures a balanced and efficient search for global optima. Moreover, the adaptation of parameters such as scaling factor S, control parameter μ, and the incorporation of randomness through Brownian motion and Lévy flight distributions further enriches HSCAGOA's exploration capabilities. These features enable the algorithm to avoid premature convergence and effectively explore diverse regions of the solution space. Empirical evaluations on a variety of benchmark test problems and engineering challenges consistently demonstrate HSCAGOA's superiority over classical SCA and other state-of-the-art optimisation algorithms. It achieves faster convergence rates, higher solution quality, and greater robustness across different optimisation landscapes, making it particularly suitable for real-world applications requiring a balance between global exploration and local exploitation. HSCAGOA emerges as a robust and versatile optimisation framework that not only mitigates the limitations of traditional SCA but also sets a new standard for addressing complex optimisation problems across various domains.

In the future, HSCAGOA is expected to demonstrate significant potential across multiple domains, particularly in addressing key challenges in machine learning, data science, financial modelling, and industrial engineering. Customising HSCAGOA for these domains and evaluating its performance against tailored benchmarks will deepen our understanding of its versatility and effectiveness. Additionally, exploring HSCAGOA's capabilities in multi-objective optimisation scenarios, where conflicting objectives require simultaneous optimisation, presents an opportunity to develop hybrid strategies that integrate multi-objective techniques. Such approaches could yield Pareto-optimal solutions, enhancing decision-making in complex environments. Furthermore, investigating hybridization with other advanced metaheuristic algorithms and machine learning methods holds promise for synergistically leveraging strengths and mitigating weaknesses, thereby advancing optimisation performance. Real-world applications, including supply chain management, renewable energy optimisation, healthcare resource allocation, and disaster management, will validate HSCAGOA's practical effectiveness. Integrating feedback from domain experts in these applications will further refine and optimise the algorithm for broader adoption and impact.

Acknowledgements This work was supported by King Saud University, Riyadh, Saudi Arabia, under Researchers Supporting Project number RSPD2025R697.

Author Contributions Gyan Singh: methodology, conceptualization, formal analysis, review and editing Saptadeep Biswas: supervision, methodology, conceptualization, formal analysis, writing—original draft, review and editing Absalom El-Shamir Ezugwu: supervision, methodology, conceptualization, writing—original draft Vladimir Simic: supervision, methodology,

 https://doi.org/10.1007/s44196-025-00823-6

conceptualization, writing—original draft Uttam Kumar Bera: supervision, methodology, conceptualization, writing—original draft Kashif Saleem: supervision, methodology, conceptualization, writing—original draft Laith Abualigah: supervision, methodology, conceptualization, writing—original draft.

Funding Open access funding provided by North-West University. This work was supported by King Saud University, Riyadh, Saudi Arabia, under Researchers Supporting Project number RSPD2025R697.

Data Availability Statements Data is available from the authors upon reasonable request. No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent During the preparation of this work, the authors utilised AI-assisted tools such as ChatGPT and Grammarly to enhance English language accuracy, including spelling, grammar, and punctuation. To ensure accuracy and originality, the authors thoroughly reviewed, revised, and edited the contents generated or corrected by these tools. The author accepts full responsibility for the originality of the final content of this publication.

References

1. Abdel-Mawgoud, H., Fathy, A., Kamel, S.: An effective hybrid approach based on arithmetic optimization algorithm and sine cosine algorithm for integrating battery energy storage system into distribution networks. J. Energy Storage **49**, 104154, 5 (2022)
2. Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M.A., Gandomi, A.H.: The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. **376**, 113609 (2021)
3. Abualigah, L., Dulaimi, A.J.: A novel feature selection method for data mining tasks using hybrid Sine Cosine Algorithm and Genetic Algorithm. Clust. Comput. **24**(3), 2161–2176, 9 (2021)
4. Abualigah, L., Elaziz, M.A., Sumari, P., Geem, Z.W., Gandomi, A.H.: Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst. Appl. **191**, 116158 (2022)
5. Agushaka, J.O., Ezugwu, A.E., Abualigah, L.: Dwarf mongoose optimization algorithm. Comput. Methods Appl. Mech. Eng. **391**, 114570 (2022)
6. Agushaka, J.O., Ezugwu, A.E., Abualigah, L.: Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. Neural Comput. Appl. **35**(5), 4099–4131, 2 (2023)
7. Ahmadianfar, I., Bozorg-Haddad, O., Chu, X.: Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf. Sci. **540**, 131–159, 11 (2020)
8. Altay, E.V., Alatas, B.: Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining. Inf. Sci. **554**, 198–221, 4 (2021)
9. Awad, N.H., Ali, M.Z., Suganthan, P.N.: Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 372–379. IEEE, 6 (2017)
10. Brajević, I., Stanimirović, P.S., Li, S., Cao, X., Khan, A.T., Kazakovtsev, L.A.: Hybrid Sine Cosine algorithm for solving engineering optimization problems. Mathematics **10**(23), 4555, 12 (2022)
11. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39, 11 (2006)
12. Duan, Y., Yu, X.: A collaboration-based hybrid GWO-SCA optimizer for engineering optimization problems. Expert Syst. Appl. **213**, 119017, 3 (2023)
13. Ezugwu, A.E., Agushaka, J.O., Abualigah, L., Mirjalili, S., Gandomi, A.H.: Prairie dog optimization algorithm. Neural Comput. Appl. **34**(22), 20017–20065, 11 (2022)

14. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**(1), 17–35, 1 (2013)
15. Gupta, S., Deep, K., Engelbrecht, A.P.: A memory guided sine cosine algorithm for global optimization. Eng. Appl. Artif. Intell. **93**, 103718, 8 (2020)
16. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18, 3 (2003)
17. Hashim, F.A., Hussain, K., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl. Intell. **51**(3), 1531–1551, 3 (2021)
18. Hashim, F.A., Hussien, A.G.: Snake optimizer: a novel meta-heuristic optimization algorithm. Knowl. Based Syst. **242**, 108320, 4 (2022)
19. Holland, J.H.: Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence, pp. 211. University of Michigan Press (1992)
20. Jamali, A.: Improving land use land cover mapping of a neural network with three optimizers of multi-verse optimizer, genetic algorithm, and derivative-free function. Egypt. J. Remote Sens. Space Sci. **24**(3), 373–390, 12 (2021)
21. Karaboga, D., Basturk, B.: Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. In: Foundations of fuzzy logic and soft computing, pp. 789–798. Springer, Berlin (2007)
22. Kaveh, A., Talatahari, S., Khodadadi, N.: Stochastic paint optimizer: theory and application in civil engineering. Eng. Comput. **38**(3), 1921–1952, 6 (2022)
23. Kennedy, J., Eberhart, R.: Particle swarm optimization. In IEEE International Conference on Neural Networks-Conference Proceedings, vol. 4, pp. 1942–1948 (1995)
24. Kumar, A., Wu, G., Ali, M.Z., Mallipeddi, R., Suganthan, P.N., Das, S.: A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol. Comput. **56**, 100693, 8 (2020)
25. Kumari, M., De, P.K., Chaudhuri, K., Narang, P.: Utilizing a hybrid metaheuristic algorithm to solve capacitated vehicle routing problem. Results Control Optim. **13**, 100292, 12 (2023)
26. Li, C., Liang, K., Chen, Y., Pan, M.: An exploitation-boosted sine cosine algorithm for global optimization. Eng. Appl. Artif. Intell. **117**, 105620, 1 (2023)
27. Li, Y., Yuan, Q., Han, M., Cui, R.: Hybrid multi-strategy improved wild horse optimizer. Adv. Intell. Syst. **4**(10), 10 (2022)
28. Liu, H., Zhang, X., Zhang, H., Li, C., Chen, Z.: A reinforcement learning-based hybrid Aquila optimizer and improved arithmetic optimization algorithm for global optimization. Expert Syst. Appl. **224**, 119898, 8 (2023)
29. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. **188**(2), 1567–1579, 5 (2007)
30. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl. Based Syst. **89**, 228–249, 11 (2015)
31. Mirjalili, S.: The Ant Lion optimizer. Adv. Eng. Softw. **83**, 80–98, 5 (2015)
32. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl. Based Syst. **96**, 120–133 (2016)
33. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**, 163–191 (2017)
34. Mirjalili, S., Lewis, A.: The Whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
35. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey Wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
36. Mohamed, A.W., Hadi, A.A., Fattouh, A.M., Jambi, K.M.: LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC), vol. 6, pp. 145–152. IEEE (2017)
37. Nadimi-Shahraki, M.H., Taghian, S., Javaheri, D., Sadiq, A.S., Khodadadi, N., Mirjalili, S.: MTV-SCA: multi-trial vector-based sine cosine algorithm. Cluster Comput. **6**, 13515 (2024)
38. Nenavath, H., Jatoth, R.K.: Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl. Soft Comput. **62**, 1019–1043, 1 (2018)
39. Nguyen, A.-T., Reiter, S., Rigo, P.: A review on simulation-based optimization methods applied to building performance analysis. Appl. Energy **113**, 1043–1058, 1 (2014)
40. Osaba, E., Villar-Rodriguez, E., Del Ser, J., Nebro, A.J., Molina, D., LaTorre, A., Suganthan, P.N., Coello, C.A., Coello, H.F.: A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. Swarm Evol. Comput. **64**, 100888 (2021)
41. Pazouki, E.: A practical surface irrigation system design based on volume balance model and multi-objective evolutionary optimization algorithms. Agric. Water Manag. **248**, 106755, 4 (2021)
42. Pham, V.H.S., Nguyen Dang, N.T., Nguyen, V.N.: Hybrid sine cosine algorithm with integrated roulette wheel selection and opposition-based learning for engineering optimization problems. Int. J. Comput. Intell. Syst. **16**(1), 171, 10 (2023)
43. Pham, V.H.S., Nguyen Dang, N.T., Nguyen, V.N.: Enhancing engineering optimization using hybrid sine cosine algorithm with Roulette wheel selection and opposition-based learning. Sci. Rep. **14**(1), 694, 1 (2024)
44. Rao, R., Savsani, V., Vakharia, D.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**(3), 303–315, 3 (2011)

 https://doi.org/10.1007/s44196-025-00823-6

45. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
46. Rather, S.A., Bala, P.S.: Hybridization of constriction coefficient-based particle swarm optimization and chaotic gravitational search algorithm for solving engineering design problems. In: Applied Soft Computing and Communication Networks Proceedings of ACN 2019, volume 125, pp. 95–115. Springer, Singapore (2020)
47. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. J. Mech. Des. Trans. ASME **112**(2) (1990)
48. Seyyedabbasi, A.: WOASCALF: a new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems. Adv. Eng. Softw. **173**, 103272, 11 (2022)
49. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
50. Singh, D., Dhillon, J.: Ameliorated grey wolf optimization for economic load dispatch problem. Energy **169**, 398–419, 2 (2019)
51. Singh, N., Kaur, J.: Hybridizing sine-cosine algorithm with harmony search strategy for optimization design problems. Soft Comput. **25**(16), 11053–11075, 8 (2021)
52. Singh, N., Singh, S.: A novel hybrid GWO-SCA approach for optimization problems. Eng. Sci. Technol. Int. J. **20**(6), 1586–1601, 12 (2017)
53. Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
54. Uzer, M.S., Inan, O.: Application of improved hybrid whale optimization algorithm to optimization problems. Neural Comput. Appl. **35**(17), 12433–12451, 6 (2023)
55. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82, 4 (1997)
56. Wu, G., Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2017)
57. Yazdani, M., Zarate, P., Kazimieras Zavadskas, E., Turskis, Z.: A combined compromise solution (CoCoSo) method for multi-criteria decision-making problems. Manag. Decis. **57**(9), 2501–2519, 10 (2019)

Authors and Affiliations

Gyan Singh[1] · Saptadeep Biswas[1] · Absalom El-Shamir Ezugwu[2] · Vladimir Simic[3,4,5] · Uttam Kumar Bera[1] · Kashif Saleem[6] · Laith Abualigah[7]

✉ Absalom El-Shamir Ezugwu
absalom.ezugwu@nwu.ac.za

✉ Laith Abualigah
aligah.2020@gmail.com

Gyan Singh
gyansingh6047@gmail.com

Saptadeep Biswas
saptadeepmath.sch@nita.ac.in

Vladimir Simic
vsima@sf.bg.ac.rs

Uttam Kumar Bera
uttam.math@nita.ac.in

Kashif Saleem
ksaleem@ksu.edu.sa

[1] Department of Mathematics, National Institute of Technology Agartala, Tripura, India

[2] Unit for Data Science and Computing, North-West University, 11 Hoffman Street, Potchefstroom 2520, South Africa

[3] Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, Belgrade 11010, Serbia

[4] Department of Industrial Engineering and Management, College of Engineering, Yuan Ze University, Taoyuan City 320315, Taiwan

[5] Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, Republic of Korea

[6] Department of Computer Science and Engineering, College of Applied Studies and Community Service, King Saud University, 11362 Riyadh, Saudi Arabia

[7] Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan