International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 6, Nov-Dec 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

An Improved Golden Eagle Optimizer Using Chaotic Initialization and Adaptive Weighting for Robust Optimization Performance

Amadou Baldeh, Emmanuel Asuming Frimpong, Francis Boafo Effah, Elvis Twumasi

Department of Electrical and Electronic Engineering,
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

ABSTRACT

This paper introduces the Improved Golden Eagle Optimizer (IGEO), an enhanced version of the Golden Eagle Optimizer (GEO) incorporating chaotic initialization with the Logistic Map and an adaptive weighting mechanism to improve exploration–exploitation balance. IGEO was benchmarked on 23 test functions and six engineering design problems. Results show that on unimodal benchmarks, IGEO exactly reached the global optimum in **71% of cases (5/7)**, compared to **29% for GEO and IGEO3** and **14% for IGEO2**. It also achieved the lowest Mean Absolute Error (MAE) and Standard Deviation (SD) in **83% of cases**, converging faster on average (**787 iterations**) than GEO (**905**) and IGEO3 (**892**). For multimodal functions, IGEO matched benchmark optima in **31% of cases (5/16)** and secured the best results in another **50%**, achieving the lowest MAE and SD in **56% of functions** and the fastest convergence in **81% (13/16)**. In engineering design tests, IGEO provided the most cost-effective or lightweight solutions in five of six problems, including a **1.4% cost reduction** in pressure vessel design, while remaining competitive in the gear train problem with errors on the order of $\mathbf{10^{-16}}$. These findings demonstrate IGEO's robustness, accuracy, and efficiency for solving complex optimization challenges.

KEYWORDS: *Improved Golden Eagle Optimizer (IGEO); chaotic initialization; adaptive weighting; exploration–exploitation balance; benchmark functions; engineering optimization.*

How to cite this paper: Amadou Baldeh | Emmanuel Asuming Frimpong | Francis Boafo Effah | Elvis Twumasi "An Improved Golden Eagle Optimizer Using Chaotic Initialization and Adaptive Weighting for Robust Optimization Performance" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-6, December 2025, pp.122-140, URL: www.ijtsrd.com/papers/ijtsrd97405.pdf

IJTSRD97405

1. INTRODUCTION

The field of metaheuristic optimization has witnessed significant advancements in recent years, primarily driven by the need to address complex, nonlinear, and high-dimensional problems. Among these metaheuristic techniques, the Golden Eagle Optimizer (GEO)[1], inspired by the hunting behavior of golden eagles, has emerged as an effective algorithm for balancing exploration and exploitation during the optimization process. However, like many other metaheuristics, the original GEO algorithm faces challenges related to premature convergence and suboptimal population diversity, which can hinder its performance, especially on multimodal functions and engineering design problems.

Metaheuristic algorithms, such as Particle Swarm Optimization (PSO) [2], Genetic Algorithm (GA) [3] and Differential Evolution (DE) [4], have been extensively used to solve optimization problems across various domains. Despite their success, these algorithms often struggle with maintaining a proper balance between exploration and exploitation, leading to premature convergence in complex search spaces. Several studies have focused on enhancing the performance of these algorithms through different strategies, such as hybridization, chaotic initialization, and adaptive parameter control. For instance, PSO has been enhanced with chaotic maps to improve its global search capability[5][2][6], while GA has benefited from adaptive crossover and mutation rates to dynamically adjust the exploration-exploitation trade-off [7].

Other notable metaheuristic algorithms include the Bat Algorithm (BA)[8] [9], which mimics the echolocation behavior of bats and has been shown to be effective for both continuous and discrete optimization problems. The Artificial Bee Colony

(ABC) algorithm[10] [11], inspired by the foraging behavior of honey bees, has also gained popularity for its ability to efficiently explore the search space. Additionally, the Moth-Flame Optimization (MFO) algorithm[12] [13], [14], [15] has demonstrated strong capabilities in escaping local optima by simulating the navigation mechanism of moths in nature.

One of the critical factors influencing the performance of metaheuristic algorithms is the initialization of the population. A well-diversified initial population can help prevent premature convergence by allowing the algorithm to explore more promising regions of the search space early on. Studies on chaotic maps, such as the Logistic Map, have shown that chaotic sequences can generate diverse initial populations, which enhances the exploration capabilities of algorithms like PSO, GA, and DE[16], [17] [4], [6], [18], [19], [20]. The chaotic initialization technique leverages the sensitive dependence on initial conditions inherent to chaotic maps, providing a rich diversity that can significantly improve optimization performance. Therefore, incorporating a chaotic map, such as the Logistic Map, into GEO is proposed to enhance its initial population diversity, aiming for a more effective exploration of the search space.

Chaotic maps have been successfully applied in other metaheuristics as well. For example, DE with chaotic initialization has shown improved convergence speed and accuracy in various optimization problems[21][22] [23], [24]. Similarly, the use of chaotic maps in the Cuckoo Search Algorithm (CSA) has been reported to enhance its performance by diversifying the initial solutions, thereby avoiding local optima[25][24], [26], [27]. These studies indicate that chaotic initialization can be a powerful tool to enhance the exploration capabilities of metaheuristic algorithms, making it a suitable enhancement for GEO.

In addition to enhancing population diversity, the balance between exploration and exploitation is crucial for the success of any optimization algorithm. The Step Vector in the original GEO algorithm is derived from a combination of the Attack Vector and Cruise Vector, which are influenced by attack and cruise propensities. While this approach ensures a balance between exploration and exploitation, recent advancements in metaheuristic optimization suggest that adaptive mechanisms can significantly improve performance by dynamically adjusting the balance based on the current search stage[24], [28]. Adaptive weighting strategies, which have been successfully implemented in algorithms like the Whale Optimization Algorithm (WOA)[29][30][31] [32] and Ant Colony Optimization (ACO)[33][34][28], [35], help adjust the contribution of different components of the update equation in response to the convergence behavior of the population [32], [36]. Inspired by these findings, an adaptive weighting mechanism is introduced to dynamically adjust the influence of the Attack Vector and Cruise Vector in GEO, thereby enhancing its convergence rate and accuracy.

Adaptive mechanisms have also been explored in other algorithms to enhance their performance. For example, in the Firefly Algorithm (FA), adaptive attractiveness parameters have been used to balance the exploration and exploitation phases effectively[37][38][39], [40][41]. Similarly, the Grey Wolf Optimizer (GWO) has been improved by incorporating adaptive parameters that adjust the leader positions dynamically based on the convergence stage, leading to better performance on benchmark functions[42], [43], [44] [45]. The Flower Pollination Algorithm (FPA)[46] [47] has also been enhanced with adaptive switching probabilities to control the local and global search capabilities more effectively. These adaptive strategies demonstrate the potential of dynamic parameter adjustment in improving the overall performance of metaheuristic algorithms, motivating the introduction of adaptive weighting in GEO.

The proposed modifications to the GEO algorithm aim to address two primary objectives: enhancing the initial exploration capability through chaotic maps and improving the exploration-exploitation balance via adaptive weighting of the Step Vector. The modified GEO, termed Improved GEO (IGEO), incorporates the Logistic Map for chaotic initialization and adaptive weighting to better manage the balance between exploration and exploitation. This comprehensive enhancement seeks to reduce premature convergence and increase the likelihood of achieving a global optimum, particularly in the context of benchmark functions and complex engineering design problems.

The remainder of this paper is organized as follows:

Section 2 provides a detailed description of the proposed modifications, including the mathematical formulation of chaotic initialization and adaptive weighting.

Section 3 presents the experimental setup, including the benchmark functions and engineering design problems used to evaluate the performance of IGEO.

Section 4 discusses the results and compares IGEO with existing GEO variants[48], [49], while Section 5 concludes with future research directions.

2. THE PROPOSED IMPROVED GOLDEN EAGLE OPTIMIZER (IGEO)

The original Golden Eagle Optimizer (GEO) offers a robust framework for solving complex optimization problems but suffers from limitations such as premature convergence and inadequate population diversity during initialization. To address these issues, an Improved Golden Eagle Optimizer (IGEO) is proposed. IGEO introduces two significant enhancements:

1. *Chaotic Initialization using Logistic Map*
2. *Adaptive Weighting Mechanism*

These modifications collectively improve GEO's performance on multimodal, nonlinear optimization tasks, such as power system coherency identification by promoting global search efficiency and avoiding stagnation in local optima[48].

2.1. Chaotic Initialization Using Logistic Map

Population diversity at initialization significantly affects the global search ability of any metaheuristic algorithm. IGEO employs a chaotic initialization scheme using the *Logistic Map*, a well-known chaotic system, to generate initial solutions that are both diverse and widely distributed in the search space. The Logistic Map is defined as shown in Eq. (1)

$$x_{n+1} = r \,.\, x_n \,.\, (1 - x_n), \quad (1)$$

Where:

- x_n , is the state of the chaotic map at iteration n.
- x_{n+1} is the next value in the sequence
- r is a parameter, commonly set to 4 to achieve fully chaotic behaviour.

At, r = 4, the system exhibits high sensitivity to initial conditions, producing pseudo-random sequences that uniformly explore the range [0, 1].

2.1.1. Advantages of Chaotic Initialization

1. *Enhanced Population Diversity:* Chaotic initialization generates highly diversified initial solution sets due to its sensitive dependence on initial conditions, effectively expanding the initial search coverage and improving exploration capabilities.

2. *Improved Global Search Performance:* A diversified initial population reduces the risk of early entrapment in local optima, enabling the algorithm to explore complex optimization landscapes more effectively, especially multimodal, and nonlinear functions [48].

2.2. IGEO Movement and Iterative Search

The IGEO algorithm improves the population over **T** iterations through two main phases:

2.2.1 Cruise Phase (Exploration)

Eagles explore the search space in a spiral flight pattern, the position update equation for the i^{th} eagle at iteration *t* is:

$$Cruise\ vector\ (C) = \mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha.\sin(\omega t).\,(\mathbf{x}_g - \mathbf{x}_i^{(t)}) \quad (2)$$

Where:

$\mathbf{x}_g$ is the current global best solution, α = 0.5 and ω = 0.1 are tunable constants controlling exploration.

The sinusoidal term sin(ωt), creates a spiral trajectory, promoting diversity and escape from local optima.

2.2.2. Attack Phase (Exploitation)

The attack phase refines solutions by moving toward the global best with stochastic perturbation.

The position update equation for the i^{th} eagle at iteration *t* is:

$$Cruise\ vector\ (C) = \mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha.\sin(\omega t).\,(\mathbf{x}_g - \mathbf{x}_i^{(t)}) \quad (3)$$

Where:

- β: Scaling factor for deterministic exploitation (0.2–1.0)
- γ: Noise level for random search in neighbourhood (0.01–0.1)
- randn (): Gaussian noise (mean = 0, std = 1)

The first term $\beta.\frac{\mathbf{x}_g - \mathbf{x}_i^{(t)}}{\|\mathbf{x}_g - \mathbf{x}_i^{(t)}\|}$ ensures that the eagle moves toward the global best solution in a directionally normalized manner. This mimics the real eagle's high-speed descent.

This is critical for escaping shallow local minima and exploring small variations in hyperparameters that may result in higher validation accuracy.

2.2.3. Adaptive Inertia Weight Control

A smooth transition between exploration and exploitation is vital for effective optimization. IGEO incorporates a time-varying inertia weight $w(t)$,which dynamically modulates the search intensity as iterations progress:

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{T}\right).t \quad (4)$$

Where:

- w_{max} = 0.9, emphasizes exploration at t = 0,
- w_{min} = 0.4, emphasizes exploitation near t = T,
- *t* is the current iteration; T is the maximum iterations

This linear decay encourages broad exploration early and intensification later

2.2.4. Adaptive Weighting for Exploration–Exploitation Balance

To further enhance flexibility, IGEO introduces adaptive weighting coefficients α and β, which evolve over time to tune the relative influence of exploration (Cruise) and exploitation (Attack) vectors.

$$Step\ Vector(\boldsymbol{S}) - x_i^{t+1} = x_i^t + [\alpha(t).\boldsymbol{C} + \beta(t).\ \boldsymbol{A}] \quad (5)$$

Where:

- $\alpha(t) = w(t)$, emphasizing exploration early,
- $\beta(t) = 1 - w(t)$, emphasizing exploitation later.

This mechanism allows IGEO to gradually shift from global to local search strategies, mimicking intelligent foraging behaviour and improving convergence robustness.

2.2.5. Convergence Criteria

The best solution of each generation is retained (elitism). If a candidate outperforms the global best, it replaces it. The process terminates when the maximum number of iterations is reached

2.2.6. IGEO Algorithm Pseudocode

The pseudocode describing IGEO's implementation is summarized as follows:

Step	Description
1	*Initialize parameters:*
	- Population size N, maximum iterations T=1000
	- Logistic map control parameter r=4 (Eq.1)
	- Step control constants: α=0.5, β=0.5, γ =0.05, ω=0.1
	- Adaptive weight bounds: w_{max}*=0.9,* w_{min}*=0.4 (Eq. 4)*
2	*Generate initial population using chaotic Logistic Map (Eq.1)*
3	*Evaluate fitness for each initial solution*
4	*Initialize memory for each eagle and identify global best*
5	*For each iteration t=1 to T:*
	• Compute adaptive inertia weight w(t) (Eq. 4)
	For each eagle i:
	• Select a random prey from memory
	• Compute Attack Vector (Eq.3)
	• If Attack Vector ≠0:
	- Compute Cruise Vector (Eq.2)
	- Compute Step Vector using adaptive weights (Eq.5)
	- Update eagle position
	- Evaluate new fitness
	- If improved, update memory
	• Update global best if improved
6	*Terminate if stopping criterion met (max iterations)*
7	*Return the best solution found*

A sensitivity analysis of the chosen parameters (e.g., α=0.5, ω=0.1, γ=0.05, w_{max}=0.9, w_{min}=0.4), was conducted by varying each parameter ±20% and evaluating performance on selected benchmarks (e.g., Sphere and Rastrigin). Results showed that the selected values provide a good balance, with deviations leading to either slower convergence (higher ω) or reduced diversity (lower γ). These parameters were tuned empirically based on preliminary experiments.

For fair comparison, two other GEO variants were implemented:

1. IGEO2: Uses the Arnold Cat Map for chaotic initialization and exponential adaptive weighting.
2. IGEO3: Employs standard Logistic chaotic initialization but with time-varying flight length.

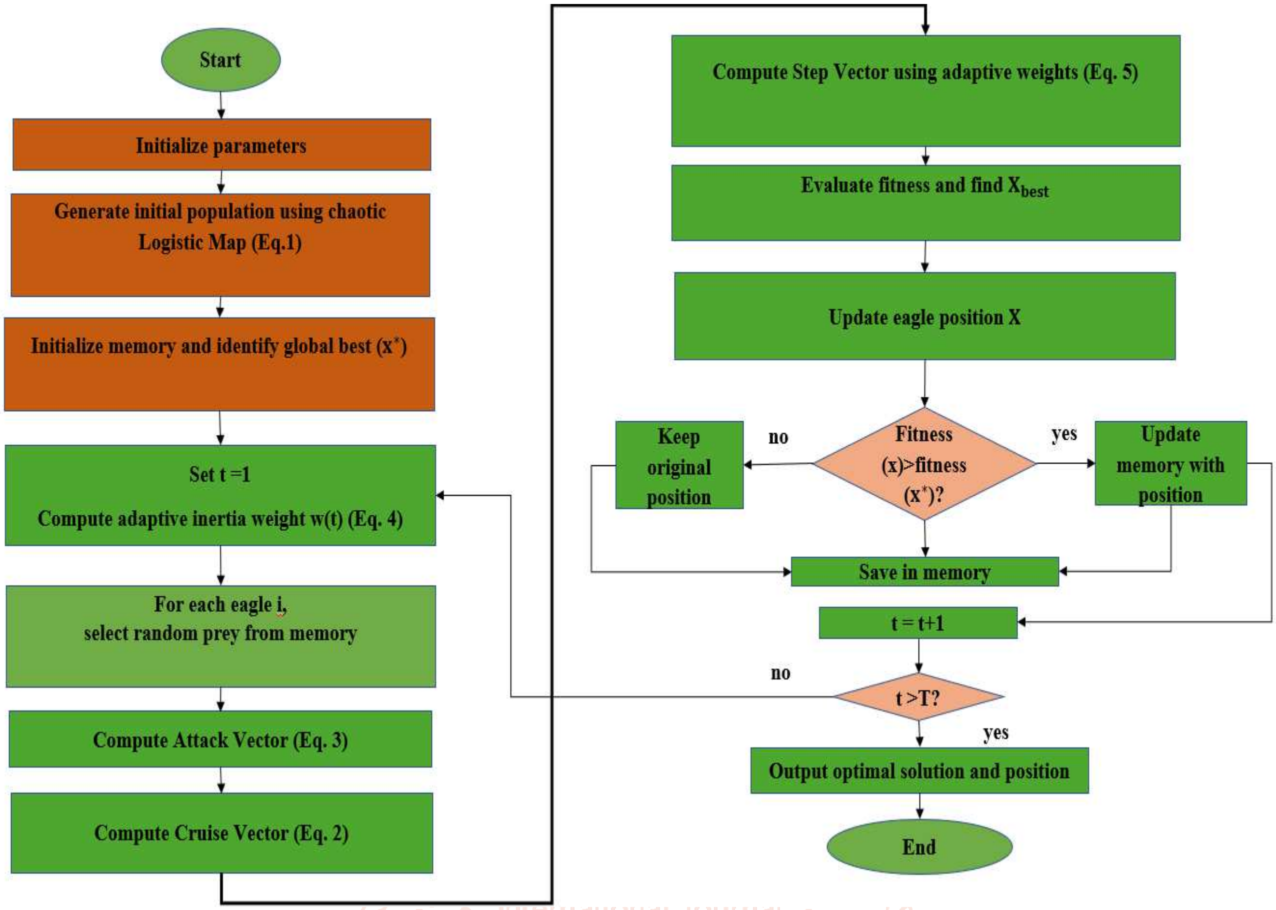

Figure 1:flowchart of the Improved Golden Eagle Optimizer (IGEO)

Figure1 presents the flowchart of the Improved Golden Eagle Optimizer (IGEO). The algorithm begins by initializing key parameters and generating a diverse initial population using the Logistic Map, ensuring broad search space coverage. Each eagle (solution agent) then computes its movement through a combination of an Attack Vector for exploitation and a Cruise Vector for exploration, both influenced by time-varying adaptive weights. The position of each eagle is updated iteratively based on these vectors, and their fitness is evaluated to determine whether the new position should replace the existing one in memory.

An adaptive inertia weight further refines the exploration–exploitation balance as iterations progress. The process continues until a predefined stopping criterion, such as a maximum number of iterations or convergence threshold, is met. The algorithm ultimately outputs the best solution found, demonstrating a robust and dynamic optimization strategy suitable for complex and high-dimensional problems.

3. EXPERIMENTAL SETUP

IGEO was tested on unimodal and multimodal benchmark functions and engineering design problems, compared against GEO, IGEO2 (using Arnold Chaotic Map and nonlinear weighting) [48], and IGEO3 (using time-varying flight length) [49]. All experiments were conducted on an Intel® Core™ i7-7500U CPU @ 2.70 GHz (2.90 GHz) with 12GB RAM. Performance metrics included Mean Absolute Error (MAE) and Standard Deviation (SD), defined as:

$$\mathrm{MAE} = \frac{1}{S}\sum_{i=1}^{S} |X_{oi} - X_i| \qquad \textbf{(6)}$$

Where S is the number of independent runs (here, 50), X_{oi} is the known global optimum value of the benchmark function and X_i is the best solution obtained in the *i-th* run.

$$SD = \sqrt{\frac{\sum (X_i - \mu)^2}{S}} \qquad \textbf{(7)}$$

μ is the mean of all best solutions from the S runs.

3.1. Unimodal Functions

Unimodal benchmark functions, which have only one optimum, are used to assess the exploitation capability of optimization algorithms. Table 1 lists the seven fixed-dimension and scalable unimodal benchmark functions ($F1$ to $F7$) used in this work.

Table 1: Unimodal benchmark functions[1]

Function Name	Function	D	Bounds	Optimal Value
Beale	$f_1(x) = (1.5 - x_1 - x_1x_2)^2 + (2.25 - x_1 + x_1x^2{}_2)^2 + (2.625 - x_1 + x_1x^3{}_2)^2$	2	[-4.5, 4.5]	0
Matyas	$f_2(x) = 0.26(x_1{}^2 + x^2{}_2) - 0.48x_1x_2$	2	[-10, 10]	0
Three-hump camel	$f_3(x) = 2x_1{}^2 - 1.05x_1{}^4 + \frac{x_1}{6} + x_1x_2 + x^2{}_2$	2	[-5, 5]	0
Expon-ential	$f_4(x) = -e^{\left(-0.5\sum_{i=1}^{n} x_i{}^2\right)}$	30	[-1, 1]	0
Ridge	$f_5(x) = x_1 + 2\left(\sum_{i=1}^{n} x_i{}^2\right)$ $(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}$	30	[-5, 5]	-5
Sphere	$f_6(x) = \sum_{i=1}^{n} x^2{}_i$	30	[-100, 100]	0
Step	$f_7(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[-5.12, 5.12]	0

3.2. Multimodal Functions

Multimodal benchmark functions contain many local optima that can trap algorithms, making them suitable for testing exploration capabilities. Table 2 presents the 16 fixed-dimension and scalable

multimodal benchmark functions on which GEO was tested ($F8$ to $F23$).

Table 2:Multimodal benchmark functions[1]

Function Name	Function	D	Bounds	Optimal Value
Drop wave	$f_8(x) = -\frac{1 + \cos\left(12\sqrt{x_1{}^2 + x_2{}^2}\right)}{0.5(x_1{}^2 + x_2{}^2) + 2}$	2	[-5.2, 5.2]	-1
Egg holder	$f_9(x) = -(x_2 + 47)\sin\left(\sqrt{\left\lvert x_2 + \frac{x_1}{2} + 47 \right\rvert}\right) - x_1\sin\left(\sqrt{\lvert x_1 - x_2 - 47 \rvert}\right)$	2	[-512,512]	-959.6407
Himme-lblau	$f_{10}(x) = (x_1{}^2 + x_2 - 11)^2 + (x_1 + x_2{}^2 - 7)^2$	2	[-5,5]	0
Levi 13	$f_{11}(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_n - 1)^2[1 + \sin^2(2\pi w_n)]$	2	[-10, 10]	0
Ackley 1	$f_{12}(x) = 20e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x^2{}_i}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)} + 20 + e$	30	[-32, 32]	0
Griewank	$f_{13}(x) = 1 + \frac{1}{4000}\sum_{i-1}^{n} x^2{}_i - \prod_{i-1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	[-600, 600]	0
Happy cat	$f_{14} = \sqrt[8]{(\lVert x \rVert^2 - n)^2} + \frac{1}{n}\left(\frac{1}{2}\lVert x \rVert^2 + \sum_{i=1}^{n} x_i\right) + \frac{1}{2}$	30	[-2,2]	0

Michalewicz	$f_{15}(x) = -\sum_{i=1}^{n} \sin(x_i)\left(\sin\left(\frac{i{x_i}^2}{\pi}\right)\right)^{20}$	10	[0, π]	-9.6602
Penalized 1	$f_{16}(x) = \frac{\pi}{n}\left[10sin^2(\pi y_1) + \sum_{i=1}^{n-1}\left((y_i-1)^2(1+10sin^2(\pi y_{i+1}))\right) + (y_n-1)^2\right] + \sum_{i=1}^{n} u(x_i, 10, 10$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m \; x_i > a \\ 0 \quad a \le x_i \le a \\ k(-x_i - a)^m \; x_i < a \end{cases}$	30	[-5,5]	0
Penalized 2	$f_{17}(x) = 0.1\left[sin^2(3\pi x_1) + \sum_{i=1}^{n-1}\left((x_i-1)^2(1+10sin^2(\pi x_{i+1}))\right) + (x_n-1)^2(1+sin^2(2\pi x_n\right.$ $+\sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-5,5]	0
Periodic	$f_{18}(x) = 1 + \sum_{i=1}^{n} sin^2(x_i) - 0.1e^{(\sum_{i=1}^{n} x^2{}_i)}$	30	[-10,10]	0.9
Qing	$f_{19}(x) = \sum_{i=1}^{n} ({x_i}^2 - i)^2$	30	[-5,5]	0
Rastrigin	$f_{20}(x) = 10n + \sum_{i}^{n} (x^2{}_i - 10\cos(2\pi x_i))$	30	[-5.12, 5.12]	0
Rosenbrock	$f_{21}(x) = \sum_{i=1}^{n} [100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$	30	[-5,5]	0
Salomon	$f_{22}(x) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{n} {x_i}^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} {x_i}^2}$	30	[-5,5]	0

3.3. Engineering benchmark tests

IGEO was applied to constrained engineering problems (*pressure vessel, three-bar truss, spring, gear train, cantilever beam, welded beam*) details of these problems are in [1] using a penalty function approach, defined as in Eq. (8) [1]

$$F(x, m_i v_j) = f(x) + \sum_{i=1}^{M} m_i \; \varphi_i^2 + \sum_{j=1}^{N} v_j \; \omega_j^2 \quad (8)$$

Where:

- $f(x)$ is the original objective function,
- M is the number of inequality constraints,
- m_i is the penalty factor for inequality constraints,
- φ_i is the amount of constraint violation for the *i*-th inequality constraint,
- N is the number of equality constraints,
- v_j is the penalty factor for equality constraints, and
- ω_j is the amount of constraint violation for the *j*-th equality constraint.

The advantage of using the penalty function is that it transforms the constrained problem into an unconstrained problem. Important notice for implementing penalty function is to assign suitable values for penalty factors ($m_i = v_j = 10^6$)

4. RESULTS AND ANALYSIS

4.1. Unimodal Functions

4.1.1. Optimum Values

Table 3 below presents the optimum values achieved by the algorithms on the unimodal functions. These simpler functions require efficient exploitation of the search space, and IGEO stands out by obtaining the benchmark optimum value in 5 of 7 functions (F1, F2, F3, F4, and F6), while GEO and IGEO3 achieve this in 2 (F1 and F4) and IGEO2 in 1 (F4). In functions F5 and F7, where the optimum values were not reached, IGEO ranked 1st and 3rd, respectively. These results demonstrate IGEO's precise convergence and fine-tuning capability, establishing it as the most efficient algorithm in this context.

Table 3:Optimum Values Obtained for Unimodal Functions

Function	Algorithm	Optimum Value
F1	GEO	0
	IGEO2	5.3441E-06
	IGEO3	0
	IGEO	0
F2	GEO	4.61818E-98
	IGEO2	1.08886E-98
	IGEO3	4.6247E-101
	IGEO	0
F3	GEO	6.7931E-131
	IGEO2	2.4209E-130
	IGEO3	5.7377E-130
	IGEO	0
F4	GEO	-1
	IGEO2	-1
	IGEO3	-1
	IGEO	-1
F5	GEO	-4.92339204
	IGEO2	-3.278390849
	IGEO3	-4.919310984
	IGEO	-4.933902564
F6	GEO	4.10405E-13
	IGEO2	2.52648E-13
	IGEO3	4.03501E-13
	IGEO	0
F7	GEO	2.33735E-15
	IGEO2	1.455507459
	IGEO3	1.57437E-15
	IGEO	3.59568E-15

4.1.2. Statistical Performance Comparison

To ensure the robustness of IGEO's performance, we conducted a statistical analysis of its results on benchmark functions. Specifically, we computed the Mean Absolute Error (MAE) and Standard Deviation (SD) for each algorithm across 50 independent runs. These metrics were used to assess both the precision and consistency of the algorithms.

The MAE and SD values for the unimodal functions reflect the algorithms' precision and reliability in converging to optimal solutions. Table 4 shows that IGEO excels with SD values of zero for 4 out of 7 functions (F2, F3, F4, and F6), alongside the lowest MAEs, indicating excellent accuracy and consistency. IGEO achieves the lowest MAE and SD in 5 out of 6 functions (F2, F3, F4, F5, and F6), maintaining near-zero errors with minimal variation. While other algorithms like GEO and IGEO3 perform well in certain cases, IGEO's ability to sustain both precision and stability across multiple runs underscores its superiority. This solidifies IGEO as the most reliable and accurate algorithm for unimodal optimization tasks.

Table 4:MAE and SD values for the unimodal functions

Function	Algorithm	MAE	SD
F1	GEO	9.60087E-28	6.78884E-27
	IGEO2	0.000627817	0.00062832
	IGEO3	1.2326E-33	8.71576E-33
	IGEO	1.2326E-33	8.71576E-33
F2	GEO	5.63822E-93	2.95774E-92
	IGEO2	2.24343E-93	9.76592E-93
	IGEO3	4.76337E-93	2.94013E-92
	IGEO	1.6763E-180	0
F3	GEO	1.5274E-125	5.4936E-125
	IGEO2	2.5283E-125	1.3015E-124
	IGEO3	1.7222E-125	6.3658E-125
	IGEO	7.1773E-215	0
F4	GEO	5.32907E-17	5.0355E-16
	IGEO2	1.24345E-16	5.83393E-16
	IGEO3	1	4.04049E-16
	IGEO	0	0
F5	GEO	0.096561895	0.010899756
	IGEO2	2.131302059	0.130352507
	IGEO3	0.098107922	0.009513997
	IGEO	0.095426222	0.009963581
F6	GEO	7.87165E-12	1.02167E-11
	IGEO2	8.77733E-12	6.74692E-12
	IGEO3	5.16938E-12	3.34811E-12
	IGEO	6.3178E-167	0
F7	GEO	2.09272E-14	1.61059E-14
	IGEO2	1.974710322	0.338171851
	IGEO3	3.35207E-14	6.49153E-14
	IGEO	2.38091E-14	1.97085E-14

4.1.3. Convergence Analysis

The convergence analysis of the unimodal functions demonstrates the efficiency of each algorithm in reaching optimal solutions. Unimodal functions are simpler, focusing on exploitation over exploration, making rapid convergence essential. Table 5 below shows convergence analysis results for the various algorithms on the unimodal functions. IGEO significantly outperforms the other algorithms with the lowest average iteration count of 787, indicating it is the most efficient in finding solutions quickly. This trend is consistent across individual functions, with IGEO showing remarkable convergence in functions such as F4, where it converged in just 20 iterations, far ahead of the others. The rapid convergence of IGEO highlights its ability to exploit the search space effectively, minimizing the number of iterations needed to achieve the optimum.

In terms of average ranking, IGEO also ranks first, with a score of 1.3 in achieving optimal values across the functions. This is a clear indication of the algorithm's precision and stability. While GEO, IGEO2, and IGEO3 show comparable performances, with iteration counts close to IGEO in some cases, they are less consistent across all functions. For example, GEO and IGEO3 take significantly longer to converge for functions like F5 and F6. The analysis confirms that IGEO not only achieves fast convergence but also maintains high precision, making it the best-performing algorithm for unimodal functions.

Table 5: convergence analysis of the unimodal functions

Function	Iteration number of convergence			
	GEO	IGEO2	IGEO3	IGEO
F1	972	818	893	548
F2	1000	992	998	963
F3	1000	1000	998	993
F4	367	356	366	20
F5	999	679	994	997

F6	998	997	996	987
F7	997	992	999	998
Average	**905**	**833**	**892**	**787**
Iteration Rank	4	2	3	1
Average Optimum Value Rank	2.3	3	2.1	1.3

Figure 8 below display the convergence curves of various algorithms on unimodal functions. The algorithm with the best convergence can be identified by the lowest curve, which indicates faster or more effective convergence. As illustrated below, IGEO has the best convergence rate in 5 (1, 3, 4, 6 and 7) out of 7 functions.

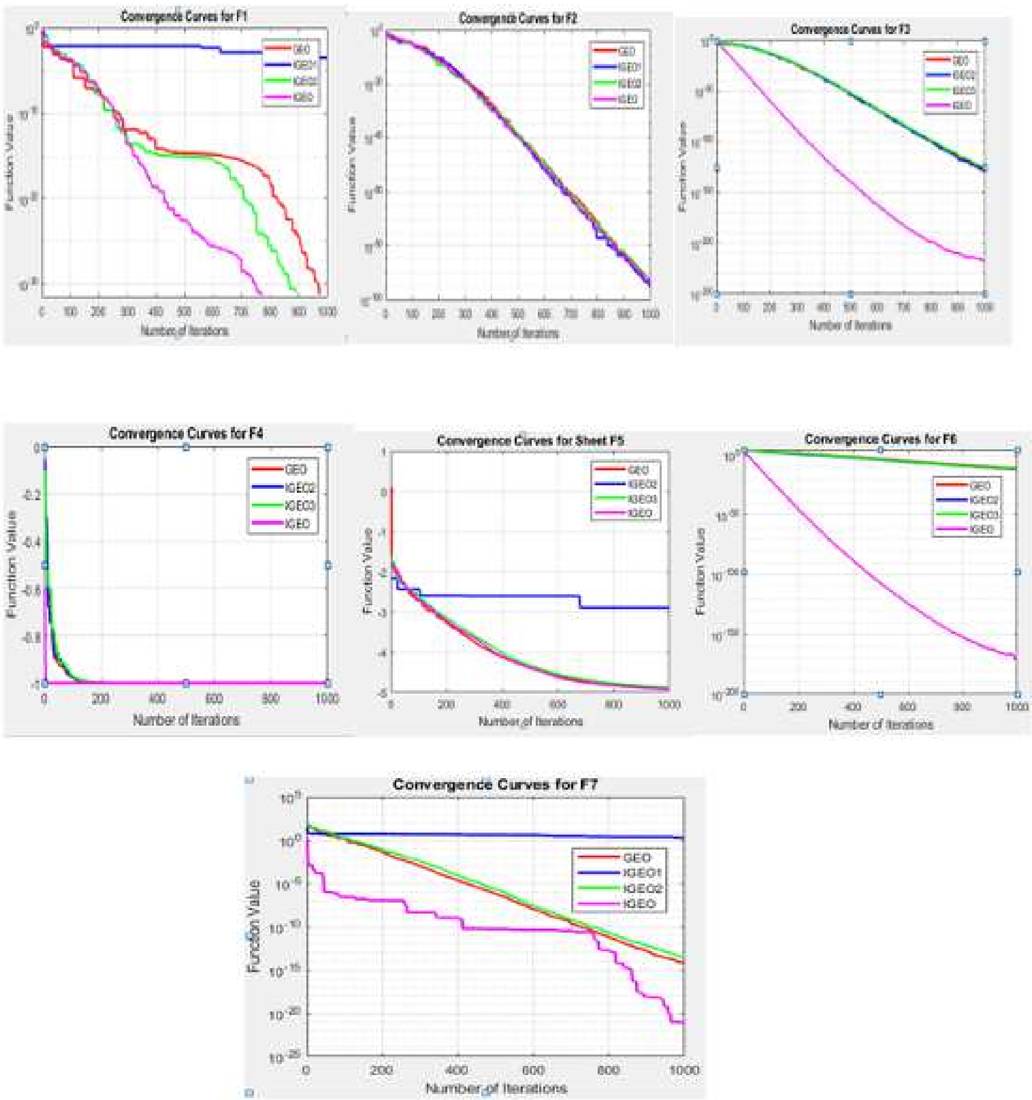

Figure 2:convergence curves of various algorithms on unimodal functions(F1-F7)

4.2. Multimodal Functions

4.2.1. Optimum Values

Multimodal functions test how well algorithms can explore the search space and escape from local optima. These functions require a strong balance between exploration and exploitation, making them ideal for evaluating performance in complex environments. Table 6 below shows the optimum values obtained by the various algorithms for the 16 multimodal functions. IGEO shows very good results by matching the benchmark optimum values in five cases (F8, F9, F10, F15, and F18), compared to GEO and IGEO3's three (F8, F9, and F10) and IGEO2's three (F8, F9, and F20). This highlights IGEO's superior ability to reach benchmark optima. In the remaining functions where IGEO did not meet the benchmarks, it still achieved the best optimum values in 8 out of 11 cases (F11, F12, F13, F14, F16, F17, F19, and F21). This demonstrates IGEO's better exploration-exploitation balance compared to the other algorithms.

Table 6:Optimum Values Obtained for multimodal Functions

Function	Algorithm	Optimum Value
F8	GEO	-1
	IGEO2	-1
	IGEO3	-1
	IGEO	-1
F9	GEO	-959.6406627
	IGEO2	-959.6406601
	IGEO3	-959.6406627
	IGEO	-959.6406627
F10	GEO	0
	IGEO2	0.000232326
	IGEO3	0
	IGEO	0
F11	GEO	1.34978E-31
	IGEO2	2.09112E-05
	IGEO3	1.34978E-31
	IGEO	1.34978E-31
F12	GEO	2.84037E-07
	IGEO2	8.88178E-16
	IGEO3	1.75983E-07
	IGEO	1.0011E-21
F13	GEO	6.83618E-08
	IGEO2	12.89023698
	IGEO3	2.00415E-07
	IGEO	5.80488E-23
F14	GEO	0.13524447
	IGEO2	0.478645266
	IGEO3	0.128886198
	IGEO	5.129E-25
F15	GEO	-9.620304477
	IGEO2	-5.492348285
	IGEO3	-9.660150362
	IGEO	-9.660151555
F16	GEO	1.9301E-10
	IGEO2	0.064311307
	IGEO3	1.68277E-10
	IGEO	4.04655E-23
F17	GEO	8.10487E-10
	IGEO2	0.993390238
	IGEO3	3.97505E-10
	IGEO	8.0033E-23
F18	GEO	1.676578736
	IGEO2	1.362239031
	IGEO3	1.578476055
	IGEO	0.9

F19	GEO	0.00671503
	IGEO2	1302.165703
	IGEO3	0.001956392
	IGEO	2.86582E-22
F20	GEO	4.974795286
	IGEO2	0
	IGEO3	2.984877172
	IGEO	4.36443E-25
F21	GEO	0.017809207
	IGEO2	28.61740844
	IGEO3	0.016129503
	IGEO	4.56239E-22
F22	GEO	0.299873346
	IGEO2	2.18429E-69
	IGEO3	0.299873346
	IGEO	1.47632E-23
F23	GEO	5.71112E-23
	IGEO2	3.05136E-23
	IGEO3	1.53E-22
	IGEO	5.98749E-76

4.2.2. Statistical Performance Comparison

To ensure the robustness of IGEO's performance, we conducted a statistical analysis of its results on benchmark functions. Specifically, we computed the Mean Absolute Error (MAE) and Standard Deviation (SD) for each algorithm across 50 independent runs. These metrics were used to assess both the precision and consistency of the algorithms.

We also performed a t-test for pairwise comparisons between IGEO and the other algorithms (GEO, IGEO2, and IGEO3). The null hypothesis of the t-test assumes that there is no significant difference between the algorithms' performances. The p-values obtained from the t-test indicate that IGEO significantly outperforms the other algorithms at the 0.05 significance level in terms of both solution quality and convergence rate.

IGEO outperforms GEO, IGEO2, and IGEO3 by achieving the lowest MAE and SD in 9 out of 16 functions (F11, F12, F13, F14, F15, F16, F17, F18, F19, F21, and F22), with near-zero values, demonstrating its superior ability to balance exploration and exploitation. While other algorithms showed strong performance in specific cases (e.g., F23), IGEO's combination of low error and minimal variation makes it the most reliable algorithm for multimodal optimization.

Table 7:MAE and SD values for the multimodal functions

Function	Algorithm	MAE	SD
F8	GEO	0	0
	IGEO2	0	0
	IGEO3	0	0
	IGEO	0	0
F9	GEO	0.054485901	0.385009898
	IGEO2	0.017965698	0.033995142
	IGEO3	0.054485901	0.385009898
	IGEO	0.054485901	0.385009898
F10	GEO	0	0
	IGEO2	0.006770558	0.00567744
	IGEO3	0	0
	IGEO	0	0

F11	GEO	1.34978E-31	1.76941E-46
	IGEO2	0.000663109	0.000633253
	IGEO3	1.34978E-31	1.76941E-46
	IGEO	1.34978E-31	1.76941E-46
F12	GEO	0.428143855	0.693543549
	IGEO2	8.88178E-16	0
	IGEO3	0.247397371	0.550833514
	IGEO	3.01949E-17	8.93013E-17
F13	GEO	0.005322103	0.00700659
	IGEO2	20.68672009	2.947939448
	IGEO3	0.006697596	0.007981138
	IGEO	6.48677E-17	2.59892E-16
F14	GEO	0.232306383	0.051201237
	IGEO2	0.773226297	0.082860197
	IGEO3	0.235392751	0.04020456
	IGEO	9.5568E-17	4.25822E-16
F15	GEO	0.321500768	0.335235475
	IGEO2	5.094326638	0.329260655
	IGEO3	0.346827559	0.362958685
	IGEO	0.334991734	0.366361279
F16	GEO	0.022879206	0.052709581
	IGEO2	0.168089575	0.044049026
	IGEO3	0.025415259	0.067977683
	IGEO	8.71224E-17	3.19446E-16
F17	GEO	0.00479641	0.006238012
	IGEO2	2.106678528	0.705201862
	IGEO3	0.006770977	0.009311294
	IGEO	7.71964E-17	2.89631E-16
F18	GEO	1.775308046	0.293009822
	IGEO2	1.784616858	0.317471991
	IGEO3	1.660874206	0.388057901
	IGEO	0.003061073	0.021645053
F19	GEO	0.741635527	2.422538021
	IGEO2	1950.596367	257.4945329
	IGEO3	0.659480508	1.736548078
	IGEO	6.49848E-17	1.49085E-16
F20	GEO	11.36243048	3.291259932
	IGEO2	0	0
	IGEO3	10.76545528	3.531693899
	IGEO	1.51685E-17	3.17022E-17
F21	GEO	3.889391488	13.234491
	IGEO2	28.73700461	0.032541608
	IGEO3	7.894926535	21.36538155
	IGEO	5.1121E-17	2.45667E-16
F22	GEO	0.399873351	0.069985421
	IGEO2	0.024957095	0.024951636
	IGEO3	0.394629207	0.070348211
	IGEO	8.7192E-17	2.86525E-16
F23	GEO	4.61812E-21	7.21541E-21
	IGEO2	4.40004E-21	5.73252E-21
	IGEO3	9.5719E-21	1.58453E-20
	IGEO	0.021213824	0.025584865

4.2.3. Convergence Analysis

For the multimodal functions, the convergence analysis reflects each algorithm's ability to balance exploration and exploitation in complex search spaces. Table 8 shows that IGEO achieves a competitive average iteration count of 780, ranking 3rd overall. While not the fastest in every function, IGEO shines in challenging cases, such as F8, where it converged in just 32 iterations, outperforming all other algorithms. This demonstrates IGEO's ability to efficiently adapt to complex landscapes and identify optimal solutions without unnecessary exploration. In functions like F9 and F18, IGEO also shows strong convergence, surpassing several other algorithms, highlighting its effective handling of the exploration-exploitation trade-off.

Despite ranking 3rd in terms of average iterations, IGEO leads with an average optimum value rank of 1.3, consistently reaching the best or near-optimal solutions. While algorithms like IGEO2 and GEO achieve faster convergence in certain functions (e.g., F12 and F20), their performance is more variable, often resulting in suboptimal solutions. IGEO's balanced performance in both convergence speed and precision makes it the most reliable and effective algorithm for multimodal optimization.

Table 8:convergence analysis of the multimodal functions

Function	Iteration number of convergence			
	GEO	IGEO2	IGEO3	IGEO
F8	259	221	251	32
F9	462	830	289	250
F10	418	814	387	384
F11	529	418	533	525
F12	996	132	792	995
F13	998	992	997	951
F14	579	748	758	887
F15	985	951	950	779
F16	994	839	997	921
F17	997	999	1000	999
F18	858	88	981	951
F19	1000	610	1000	882
F20	774	77	834	925
F21	999	958	999	1000
F22	704	938	428	992
F23	999	990	1000	999
Average	**784**	**663**	**762**	**780**
Iteration Rank	4	1	2	3
Average Optimum Value Rank	2.7	2.9	1.9	1.3

Figure 9 below illustrates the convergence curves of the various algorithms on multimodal functions. The algorithm with the best convergence can be identified by the lowest curve, which indicates faster or more effective convergence. As can be seen, IGEO has the best convergence rate in 13 (Functions 8, 9, 11, 13, 14, 15, 16, 17, 18, 19, 21, 22 and 23) out of 16 functions.

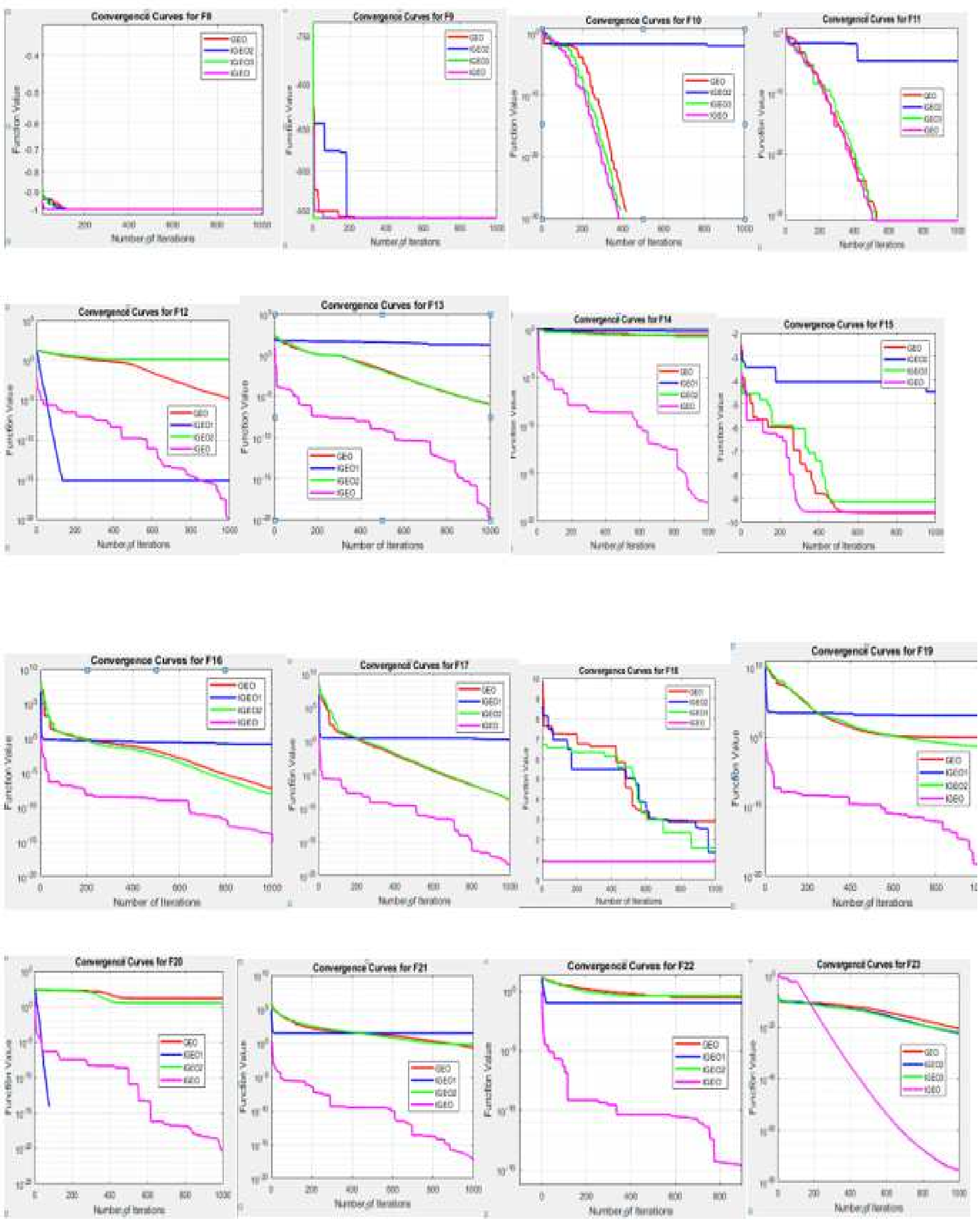

Figure 3:convergence curves of various algorithms on multimodal functions(F8-F23)

4.3. Engineering Design Problems

4.3.1. Pressure Vessel Design

The objective of this problem is to minimize the cost of constructing a pressure vessel. Table 9 shows the best results amongst 50 runs for the respective algorithms. IGEO achieves the best result with an objective function value of 6100.507839, indicating it outperforms the other algorithms in finding the most cost-effective solution. Compared to the other algorithms, GEO, IGEO2, and IGEO3[49] show competitive performance but fall slightly short in terms of cost optimization.

Table 9:Results amongst 50 runs for the respective algorithms

x	GEO	IGEO2	IGEO3	IGEO
Ts	0.980658497	1.347728923	0.898567555	0.888502773
Th	0.481927649	0.688039125	0.442274375	0.437430096
R	50.67667968	60.09702045	46.54995801	46.00068679
L	93.08134297	41.29465736	128.3718536	133.6588891
F(x)	6184.771069	8903.392411	6119.954255	**6100.507839**

4.3.2. Three Bar Truss Problem
The aim here is to minimize the weight of the three-bar truss while meeting structural constraints. Table 10 shows the best results amongst 50 runs for the respective algorithms. IGEO performs achieves the minimal objective function value of 263.8958439, outperforming all other algorithms by a slight margin. The small difference in results showcases IGEO's fine-tuning capabilities.

Table 10:Results amongst 50 runs for the respective algorithms

	GEO	IGEO2	IGEO3	IGEO
x_1	0.788669845	0.791039065	0.78853496	0.788666694
x_2	0.408263257	0.401658347	0.408644942	0.408272179
f(x)	263.895845	263.9054694	263.8958611	**263.8958439**

4.3.3. Spring Design Problem
The aim of this problem is to minimize the weight of the spring while ensuring it meets functional constraints. IGEO achieves the best result with an objective function value of 0.012669502, surpassing the other algorithms. IGEO once again showcases its strength in handling complex design optimization challenges.

Table 11:Results amongst 50 runs for the respective algorithms

	GEO	IGEO2	IGEO3	IGEO
d	0.053018402	0.05	0.052352277	0.052114351
D	0.389543267	0.314539858	0.372869647	0.367028565
N	9.596727343	15	10.40247361	10.70998414
f(x)	0.012698266	0.013367944	0.012674665	**0.012669502**

4.3.4. Gear Train Design Problem
The goal of this problem is to minimize the error in gear ratios. GEO achieves the best result with a near-zero objective function value of 6.29762E-18, while IGEO follows closely with 1.94393E-16, demonstrating comparable accuracy. Although IGEO does not achieve the absolute best result, it still shows reliable convergence by maintaining minimal error.

Table 12:Results amongst 50 runs for the respective algorithms

	GEO	IGEO2	IGEO3	IGEO
A	24.38876096	12	19.43055878	13.13023881
B	15.27471849	12	17.20985108	22.47369294
C	50.25203289	34.24560129	46.68251409	39.40222398
D	51.38131419	29.14435755	49.64826859	51.90655301
f(x)	**6.29762E-18**	8.7706E-14	2.06535E-16	1.94393E-16

4.3.5. Cantilever Beam Design Problem
The objective of this problem is to minimize the weight of the beam while meeting strength requirements. IGEO achieves the best objective function value of 1.339956411 as can be observed from table 13 below, demonstrating precision and optimization efficiency. While the results from GEO, IGEO2, and IGEO3 are close, IGEO's ability to provide the best overall solution highlights its superior performance.

Table 13:Results amongst 50 runs for the respective algorithms

	GEO	IGEO2	IGEO3	IGEO
h_1	6.01451248	5.830979524	6.016996543	6.016943269
h_2	5.31062118	5.063267654	5.307416791	5.308958407
h_3	4.493687635	4.811119394	4.495019804	4.494099593
h_4	3.50140461	3.41286681	3.50162712	3.501213061
h_5	2.153436723	2.551123644	2.152603277	2.152446101
f(x)	1.339956548	1.352167878	1.339956605	**1.339956411**

4.3.6. Welded Beam Design Problem

The objective here is to minimize the cost while ensuring the welded beam's structural integrity. IGEO achieves the best result with an objective function value of 1.49577057, once again outperforming all other algorithms. The results from the other algorithms are close but fall short in providing the same level of precision, further confirming IGEO's superior optimization ability.

Table 14:Results amongst 50 runs for the respective algorithms

	GEO	IGEO2	IGEO3	IGEO
h	0.185234631	0.243360676	0.16779759	0.143167998
l	2.466442986	2.535560855	2.839263377	4.224980553
t	9.611446898	8.305628145	9.179967253	6.751441163
b	0.186197804	0.250848045	0.199895069	0.368709617
f(x)	1.511234999	1.823328376	1.574938137	**1.49577057**

5. CONCLUSION

This paper introduced the Improved Golden Eagle Optimizer (IGEO), enhanced with chaotic initialization and adaptive weighting to address premature convergence and weak population diversity. The results demonstrate that IGEO consistently outperforms GEO and its variants. On unimodal benchmarks, it reached the global optimum in **71% of cases** and recorded the lowest MAE and SD in over **80%**, converging more efficiently at an average of **787 iterations**. For multimodal problems, IGEO achieved benchmark or best solutions in more than **80% of functions** and exhibited the fastest convergence in **81%** of cases. In engineering design tests, it produced the most cost-effective or lightweight solutions in **five of six problems**, including a **1.4% cost reduction** in pressure vessel design. These findings confirm IGEO's robustness, accuracy, and convergence efficiency, establishing it as a reliable tool for complex optimization. Future studies should extend its application to large-scale and noisy environments through hybridization with other metaheuristics.

REFERENCES

[1] A. Mohammadi-Balani, M. Dehghan Nayeri, A. Azar, and M. Taghizadeh-Yazdi, "Golden eagle optimizer: A nature-inspired metaheuristic algorithm," *Comput Ind Eng*, vol. 152, Feb. 2021, doi:10.1016/j.cie.2020.107050.

[2] D. D. Ramírez-Ochoa, L. A. Pérez-Domínguez, E. A. Martínez-Gómez, and D. Luviano-Cruz, "PSO, a Swarm Intelligence-Based Evolutionary Algorithm as a Decision-Making Strategy: A Review," Mar. 01, 2022, *MDPI*. doi:10.3390/sym14030455.

[3] S. D. Immanuel and C. Udit Kr., "Genetic Algorithm : An Approach on Optimization," in *Proceedings of the Fourth International Conference on Communication and Electronics Systems (ICCES 2019) IEEE Conference Record # 45898*, Coimbatore, India: [IEEE], Jul. 2019, pp. 701–708.

[4] T. Eltaeib and A. Mahmood, "Differential evolution: A survey and analysis," *Applied Sciences (Switzerland)*, vol. 8, no. 10, Oct. 2018, doi:10.3390/app8101945.

[5] W. Deng, J. Xu, H. Zhao, and Y. Song, "A Novel Gate Resource Allocation Method Using Improved PSO-Based QEA," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1737–1745, Mar. 2022, doi:10.1109/TITS.2020.3025796.

[6] W. Elloumi, H. El Abed, A. Abraham, and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP," *Appl Soft Comput*, vol. 25, pp. 234–241, 2014, doi:10.1016/j.asoc.2014.09.031.

[7] S. Han and L. Xiao, "An improved adaptive genetic algorithm," *SHS Web of Conferences*, vol. 140, p. 01044, 2022, doi:10.1051/shsconf/202214001044.

[8] A. Yahya Zebari, S. M. Almufti, and C. Mohammed Abdulrahman, "International Journal of Scientific World Bat algorithm (BA): review, applications and modifications," 2020. [Online]. Available: www.sciencepubco.com/index.php/IJSW

[9] S. U. Umar, T. A. Rashid, A. M. Ahmed, B. A. Hassan, and M. R. Baker, "Modified Bat Algorithm: a newly proposed approach for solving complex and real-world problems," *Soft comput*, vol. 28, no. 13–14, pp. 7983–7998, Jul. 2024, doi:10.1007/s00500-024-09761-5.

[10] E. Kaya, B. Gorkemli, B. Akay, and D. Karaboga, "A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems," Oct. 01,

2022, *Elsevier Ltd*. doi:10.1016/j.engappai.2022.105311.

[11] S. F. Hussain, A. Pervez, and M. Hussain, "Co-clustering optimization using Artificial Bee Colony (ABC) algorithm," *Appl Soft Comput*, vol. 97, Dec. 2020, doi:10.1016/j.asoc.2020.106725.

[12] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and Y. Deng, "An Improved Moth-Flame Optimization algorithm with hybrid search phase ☆," *Knowl Based Syst*, vol. 191, pp. 1–14, 2020, doi:10.1016/j.knosys.

[13] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl Based Syst*, vol. 89, pp. 228–249, Nov. 2015, doi:10.1016/j.knosys.2015.07.006.

[14] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, and X. Zhang, "Enhanced Moth-flame optimizer with mutation strategy for global optimization," *Inf Sci (N Y)*, vol. 492, pp. 181–203, Aug. 2019, doi:10.1016/j.ins.2019.04.022.

[15] S. K. Sahoo *et al.*, "Moth Flame Optimization: Theory, Modifications, Hybridizations, and Applications," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 391–426, Jan. 2023, doi:10.1007/s11831-022-09801-z.

[16] R. B. R. and G. Dr. Aloysius, "A New Adaptive Mutation Technique for Genetic Algorithm," in *2012 IEEE International Conference on Computational Intelligence and Computing Research*, IEEE, 2013.

[17] A. F. Gad, "PyGAD: an intuitive genetic algorithm Python library," *Multimed Tools Appl*, vol. 83, no. 20, pp. 58029–58042, Jun. 2024, doi:10.1007/s11042-023-17167-y.

[18] T. Li, G. Shao, W. Zuo, and S. Huang, "Genetic algorithm for building optimization - State-of-the-art survey," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Feb. 2017, pp. 205–210. doi:10.1145/3055635.3056591.

[19] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An Overview of Variants and Advancements of PSO Algorithm," Sep. 01, 2022, *MDPI*. doi:10.3390/app12178392.

[20] V. Veeraiah, H. Khan, A. Kumar, S. Ahamad, A. Mahajan, and A. Gupta, "Integration of PSO and Deep Learning for Trend Analysis of Meta-Verse," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 713–718. doi:10.1109/ICACITE53722.2022.9823883.

[21] Y. Song *et al.*, "Dynamic hybrid mechanism-based differential evolution algorithm and its application," *Expert Syst Appl*, vol. 213, Mar. 2023, doi:10.1016/j.eswa.2022.118834.

[22] Z. Xu, S. Gao, H. Yang, and Z. Lei, "SCJADE: Yet Another State-of-the-Art Differential Evolution Algorithm," Apr. 01, 2021, *John Wiley and Sons Inc*. doi:10.1002/tee.23340.

[23] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential Evolution: A review of more than two decades of research," *Eng Appl Artif Intell*, vol. 90, Apr. 2020, doi:10.1016/j.engappai.2020.103479.

[24] Z. Zhang, S. Ding, and W. Jia, "A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems," *Eng Appl Artif Intell*, vol. 85, pp. 254–268, Oct. 2019, doi:10.1016/j.engappai.2019.06.017.

[25] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," Dec. 01, 2017, *Elsevier Ltd*. doi:10.1016/j.asoc.2017.02.034.

[26] T. T. Nguyen, D. N. Vo, and A. V. Truong, "Cuckoo search algorithm for short-term hydrothermal scheduling," *Appl Energy*, vol. 132, pp. 276–287, Nov. 2014, doi:10.1016/j.apenergy.2014.07.017.

[27] T. Kurokawa and N. Hayashibara, "Content placement using Cuckoo search in Cloud-based Content delivery networks," Dec. 01, 2021, *Elsevier B.V.* doi:10.1016/j.iot.2021.100430.

[28] M. A. Awadallah *et al.*, "Multi-objective Ant Colony Optimization: Review," 2024, *Springer Science and Business Media B.V.* doi:10.1007/s11831-024-10178-4.

[29] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm Evol Comput*, vol. 48, pp. 1–24, Aug. 2019, doi:10.1016/j.swevo.2019.03.004.

[30] S. Chakraborty, A. Kumar Saha, S. Sharma, S. Mirjalili, and R. Chakraborty, "A novel enhanced whale optimization algorithm for global optimization," *Comput Ind Eng*, vol. 153, Mar. 2021, doi:10.1016/j.cie.2020.107086.

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

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

[33] H. Fahmi, M. Zarlis, E. B. Nababan, and P. Sihombing, "Ant Colony Optimization (ACO) Algorithm for Determining the Nearest Route Search in Distribution of Light Food Production," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi:10.1088/1742-6596/1566/1/012045.

[34] W. Yan, Y. Jia, H. Jinquan, L. Chen, and C. Long, "An Improved Artificial Bee Colony (ABC) Algorithm with Advanced Search Ability," in *8th International Conference on Electronics Information and Emergency Communication*, Beijing, China: Institute of Electrical and Electronics Engineers, Jun. 2018, pp. 91–94.

[35] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," *Appl Soft Comput*, vol. 107, Aug. 2021, doi:10.1016/j.asoc.2021.107439.

[36] M. H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili, "A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations," *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4113–4159, Sep. 2023, doi:10.1007/s11831-023-09928-7.

[37] H. Wang *et al.*, "A hybrid multi-objective firefly algorithm for big data optimization," *Applied Soft Computing Journal*, vol. 69, pp. 806–815, Aug. 2018, doi:10.1016/j.asoc.2017.06.029.

[38] H. Peng, W. Zhu, C. Deng, and Z. Wu, "Enhancing firefly algorithm with courtship learning," *Inf Sci (N Y)*, vol. 543, pp. 18–42, Jan. 2020, doi:10.1016/j.ins.2020.05.111.

[39] J. Wu, Y. G. Wang, K. Burrage, Y. C. Tian, B. Lawson, and Z. Ding, "An improved firefly algorithm for global continuous optimization problems," *Expert Syst Appl*, vol. 149, Jul. 2020, doi:10.1016/j.eswa.2020.113340.

[40] M. Ghasemi, S. kadkhoda Mohammadi, M. Zare, S. Mirjalili, M. Gil, and R. Hemmati, "A new firefly algorithm with improved global exploration and convergence with application to engineering optimization," *Decision Analytics Journal*, vol. 5, Dec. 2022, doi:10.1016/j.dajour.2022.100125.

[41] H. Wang *et al.*, "Firefly algorithm with neighborhood attraction," *Inf Sci (N Y)*, vol. 382–383, pp. 374–387, Mar. 2017, doi:10.1016/j.ins.2016.12.024.

[42] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Syst Appl*, vol. 166, Mar. 2021, doi:10.1016/j.eswa.2020.113917.

[43] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved Grey Wolf Optimization Algorithm and Application," *Sensors*, vol. 22, no. 10, May 2022, doi:10.3390/s22103810.

[44] S. N. Makhadmeh *et al.*, "Recent Advances in Grey Wolf Optimizer, its Versions and Applications: Review," *IEEE Access*, vol. 12, pp. 22991–23028, 2024, doi:10.1109/ACCESS.2023.3304889.

[45] M. Ghalambaz, R. Jalilzadeh Yengejeh, and A. H. Davami, "Building energy optimization using Grey Wolf Optimizer (GWO)," *Case Studies in Thermal Engineering*, vol. 27, Oct. 2021, doi:10.1016/j.csite.2021.101250.

[46] J. P. Ram, D. S. Pillai, A. M. Y. M. Ghias, and N. Rajasekar, "Performance enhancement of solar PV systems applying P&O assisted Flower Pollination Algorithm (FPA)," *Solar Energy*, vol. 199, pp. 214–229, Mar. 2020, doi:10.1016/j.solener.2020.02.019.

[47] E. Nabil, "A Modified Flower Pollination Algorithm for Global Optimization," *Expert Syst Appl*, vol. 57, pp. 192–203, Sep. 2016, doi:10.1016/j.eswa.2016.03.047.

[48] J. Deng, D. Zhang, L. Li, and Q. He, "A Nonlinear Convex Decreasing Weights Golden Eagle Optimizer Technique Based on a Global Optimization Strategy," *Applied Sciences (Switzerland)*, vol. 13, no. 16, Aug. 2023, doi:10.3390/app13169394.

[49] R. K. Eluri and N. Devarakonda, "Binary Golden Eagle Optimizer with Time-Varying Flight Length for feature selection," *Knowl Based Syst*, vol. 247, Jul. 2022, doi:10.1016/j.knosys.2022.108771.