Decision Science Letters 14 (2025) ***–***

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

An improved pelican optimization algorithm for function optimization and constrained engineering design problems

Haval Tariq Sadeeq[a*], Araz Abrahim[a], Thamer Hameed[b], Najdavan Kako[a], Reber Mohammed[c] and Dindar Ahmed[a]

[a]*Information Technology Department, Technical College of Duhok, Duhok Polytechnic University, Duhok 42001, Kurdistan Region of Iraq*
[b]*Basic Sciences Department, Agriculture College, University of Duhok, Duhok 42001, Kurdistan Region of Iraq*
[c]*Computer Science Department, College of Science, University of Zakho, Duhok 42001, Kurdistan Region of Iraq*

C H R O N I C L E

Article history:
Received: February 7, 2025
Received in revised format: March 12, 2025
Accepted: April 5 2025
Available online: April 5 2025

Keywords:
Metaheuristic algorithms
Engineering optimization
Constrained design problems
Pelican optimization algorithm
Improved pelican optimization algorithm

A B S T R A C T

Metaheuristic algorithms are a class of optimization techniques that have revolutionized problem-solving across various domains. These algorithms provide a versatile and powerful approach to finding near-optimal solutions for complex, combinatorial, and computationally intensive problems. They draw inspiration from natural processes, such as evolution, swarm behavior, or annealing, to iteratively refine solutions by intelligently navigating the problem space. Metaheuristics have become indispensable tools in both academia and industry, helping researchers and practitioners address real-world problems efficiently and effectively. The Pelican optimization algorithm (POA) is a recently developed metaheuristic algorithm that simulates the hunting behavior of pelicans. In complex optimization problems, an POA may have slow convergence or fall in sub-optimal regions, especially in high complex ones. In this paper, Levy flight is integrated into the exploration phase to enhance its search capabilities. Furthermore, a novel exponential parameter has been introduced to enhance the algorithm's overall performance by facilitating a smoother shift between exploration and exploitation phases. These modifications are intended to keep the algorithm from being locked in local optima. The developed algorithm named as IPOA was tested using widely recognized twenty-three benchmark functions with a variety of characteristics, a set of CEC2022 test suites, and five different engineering constrained problems. The results demonstrate the superiority and effectiveness of IPOA in tackling function optimization and constrained design engineering problems.

1. Introduction

In practical scenarios, optimization problems are often treated as a "black box" model, consisting of three primary components: input, model, and output. In cases where both the model and the desired output are known, and the objective is to find the appropriate input, this scenario is referred to as an optimization problem (Saleem & Gallagher, 2022). Examples of optimization problems include feature selection, and scheduling. Conversely, when certain inputs and models are known, and the objective is to input these conditions into the model to determine the corresponding output, this situation is termed a simulation problem. Simulation problems find applications in engineering design, particularly in predicting scenarios (Le Digabel & Wild, 2023). Typically, optimization problems involve a set of design variables utilized as inputs for an objective function, which acts as a model where the desired outcome is either known or can be quantified. The primary goal is to pinpoint the optimal values for these design variables, either minimizing or maximizing the objective function. Depending on the value ranges of these variables, the various

* Corresponding author.
E-mail address haval.tariq@dpu.edu.krd (H. T. Sadeeq)

ISSN 1929-5812 (Online) - ISSN 1929-5804 (Print)

doi: 10.5267/j.dsl.2025.4.004

combinations of decision variables create an extensive search space. The definition of this search space cannot be separated to the distinctive features of the scenario at consideration. The difficulty of these problems is commonly determined by examining the ruggedness of the search space and the dimensions of the solution. The ruggedness of the search space is typically influenced by the constraints of the problem, while solution dimensions are often related to the problem's scale (Gandomi & Deb, 2020).

The utilization of optimization algorithms to tackle a multitude of intricate optimization problems has experienced a notable surge in recent times. Prior to this advancement, mathematical methods like dynamic, linear, and nonlinear programming were employed to handle complex optimization challenges (Wang et al., 2024; Ajagekar et al., 2022; Zhong et al., 2024). While these approaches are proficient at obtaining optimal solutions, they are not applicable to a broad spectrum of nondeterministic polynomial-time complete problems. These are problems for which determining an exact solution within polynomial time is infeasible, and the time required increases exponentially with the input size. Consequently, these methods are not suitable for practical real-world applications (Mataifa et al., 2022).

To overcome these limitations, metaheuristic optimization algorithms have been introduced such as Giant trevally optimizer (Sadeeq & Abdulazeez, 2022a), Chernobyl disaster optimizer (Shehadeh, 2023), Artificial hummingbird algorithm (Zhao et al., 2022), elk herd optimizer (Al-Betar et al., 2024) and Leopard seal optimization (Rabie et al., 2023). These algorithms, inspired by the observation and simulation of intelligent behaviors and natural processes, offer effective solutions to complex optimization problems in polynomial time, particularly when dealing with extensive problems. Moreover, they address a significant issue encountered by local search algorithms, which is the tendency for search agents to become stuck in local regions far from the desired global solution area. Metaheuristic algorithms are primarily designed to prevent such confinement to local optima. They achieve global optima through the use of intelligent stochastic operators that explore the entire search space. Therefore, the overall search performance of metaheuristic algorithms hinges on achieving a suitable balance between exploration and exploitation (Sadeeq & Abdulazeez, 2023b; Kuang et al., 2024).

Exploration and exploitation are fundamental concepts in optimization algorithms. Exploration refers to the strategy of diversifying the search by exploring various regions of the solution space, often through randomization. It aims to discover new and potentially better solutions. On the other hand, exploitation involves intensively examining the current best-known solutions or their current neighborhoods to refine and improve them (Sadeeq & Abdulazeez, 2023b). Finding the correct balance between exploration and exploitation becomes essential for optimization algorithms. Excessive exploration may lead to slow convergence, while excessive exploitation may cause the algorithm to become stuck in local optima. Finding the optimal trade-off between these two aspects is a key challenge in algorithm design to successfully navigate through the solution domain and arrive at superior solutions (Sadeeq & Abdulazeez, 2022b).

Although numerous metaheuristic algorithms are available in the existing literature, the continuous development of new metaheuristics or enhancements to current ones for addressing various problem domains such as global optimization (Abu-Hashem & Shambour, 2024), feature selection (Houssein et al., 2023), data classification (Amine Tahiri et al., 2023; Zhong et al., 2023), feature extraction (Yang et al., 2023), image segmentation (Emam et al., 2023), scheduling (Wan et al., 2020), electronic circuits (Wongvanich et al., 2023), data regression domain problems (Yu et al., 2024; Latifi Amoghin et al., 2024), path planning (Dao et al., 2024; Tian et al., 2024) and power systems (Hashish et al., 2023; Alghamdi, 2024) remains a vibrant area of research. This necessity is underscored by the No Free Lunch (NFL) theorem, which asserts that there isn't a single metaheuristic universally suited for solving all optimization problems (Wolpert & Macready, 1997). Consequently, the choice to improve the Pelican Optimization Algorithm (POA) proposed by Trojovsky and Dehghani (2022) stems from its inherent limitations, such as a tendency to stagnate in local optima and slow convergence, particularly for complex and high-dimensional problems. These challenges hinder its performance in obtaining global optima. The motivation behind this research is to address these inefficiencies by introducing mechanisms like Lévy flights and an exponential parameter. These enhancements improve the exploration and exploitation balance, prevent premature convergence, and enhance global search capabilities. This makes the improved POA (IPOA) more robust and effective for diverse optimization and engineering design problems.

Researchers have developed various versions of POA algorithms for many kinds of tasks in computer science. For instance, (Kusuma & Prasasti, 2022) proposed a guided pelican algorithm (GPA) by making some improvements to the original algorithm. Then, GPA is employed to optimize several benchmark functions and is compared against various metaheuristic algorithms. The outcomes demonstrate that GPA surpasses all other algorithms in optimizing most benchmark functions.

To address building energy optimization challenges, a hybrid approach combining the POA and the single candidate optimizer (SCO) has been introduced in (Yuan et al., 2023). This hybrid method, known as POSCO, leverages the global search capabilities of pelican optimization along with the local search capabilities of the single candidate method. POSCO has been employed to minimize the annual energy consumption of office buildings, resulting in noticeable energy savings compared to both POSCO and POA procedures.

Authors in (Mohammed et al., 2022) have used POA for the trial-and-error hyperparameter tuning for their proposed optimal hybrid deep belief network method in smart cities. It was shown that POA improves the general effectiveness of the circulation prediction mechanism. In yet another attempt (Alamir et al., 2023), proposed a new utilization of POA to address the multi-objective optimization challenge of energy management in a microgrid. This approach takes into account a hybrid demand response program. The primary goal of this energy management problem is to minimize the total operational expenses, which encompass both generation costs and transaction costs, while simultaneously maximizing the benefits for the microgrid operator.

The authors of (Parvathi et al., 2022) have utilized POA to minimize the amount of load to be shed by determining the ideal amount of load to be shed within the restrictions provided by customers. Various kinds of distributed and renewable power generators are modeled in IEEE 33-bus simulations. The computational performance of POA is analyzed and compared to other methods with 25 independent runs per method. It has been noted that the findings produced by POA are superior in terms of execution time and global optima.

The main objective of this paper is to improve the standard POA by proposing the IPOA and applying the IPOA approach for solving global optimization benchmarks and engineering design optimization problems. Considering the modifications introduced to the original POA as outlined in this paper, the key contributions of this study can be summed up as follows:

- The introduction of the IPOA algorithm, which incorporates a novel transition strategy and new random positioning to enhance exploration and exploitation capabilities, achieve an optimal balance between exploration and exploitation, and accelerate convergence.

- The IPOA's performance is evaluated through benchmarking against a set of standard optimization benchmark functions. Furthermore, The IPOA is implemented and evaluated under the new CEC2022 benchmarks. The superiority and consistency of the IPOA method over baseline algorithms is further demonstrated by the observations and statistical data.

- The performance of IPOA is assessed across five engineering design problems: Tension/Compression Spring Design, Tubular Column Design Optimization Problem, Speed Reducer Design, Welded Beam Design, and I-Shaped Beam Design.

In conclusion, the IPOA offers excellent results to global optimization and constrained optimization problems.

The rest of this paper is structured as follows: Section 2 provides a concise overview of POA, featuring mathematical operations. Section 3 outlines the methodology and components of the proposed IPOA. Section 4 focuses on performance evaluation and results discussion, involving the application of the IPOA algorithm to global optimization benchmarks. Section 5 presents the application of IPOA on different constrained engineering problems. Section 6 concludes the paper.

2. Pelican Optimization Algorithm Background Principle

The Pelican Optimization Algorithm (POA) is a novel stochastic optimization method inspired by the foraging strategy employed by pelicans (Trojovský & Dehghani, 2022). Pelicans are known to hunt in coordinated groups. When they spot their prey in the water, pelicans dive and then extend their wings to corral the prey towards the water's surface and shallower areas, making it easier to capture. The POA algorithm is structured to mimic this behavior and can be outlined in the following manner:

2.1. Phase 1: Moving Towards Hunting Locations (Exploration Phase)

In this stage, the algorithm replicates the pelican's method of locating food sources through a search process across the designated area. Like pelicans moving towards their prey area to detect food sources, POA candidates initiate a search for their prey's location. The prey's location is determined randomly within the POA. The mathematical representation of the updated status of the *ith* pelican candidate during this phase is illustrated in Eq. (1):

$$x_{(i,j)}^{new,p1} = \begin{cases} x_{i,j} + r\left(P_{i,j} - I \times x_{i,j}\right), F_{pi} < F_i \\ x_{i,j} + r\left(x_{i,j} - P_{i,j}\right), \quad F_{pi} \geq F_i \end{cases} \tag{1}$$

where Pi is the *ith* pelican's prey position, Fpi is its objective function value, F_i is the current best objective function, I is a randomly chosen number, which can take on a value of either 1 or 2. r is a random number between [0, 1]. Equation 1 indicates that if the fitness value of the randomly chosen pelican $P_{i,j}$ is superior to that of the current pelican $x_{i,j}$, the current pelican adjusts its position by moving towards the randomly selected pelican. Conversely, if the fitness value of $P_{i,j}$ is not better, the current pelican increases its distance from $P_{i,j}$.

Within POA, the acceptance of a new pelican position depends on whether it leads to an enhancement in the value of the objective function. This process prevents the algorithm from venturing into suboptimal areas. Equation (2) serves as the mathematical representation of this mechanism:

$$X_i = \begin{cases} X_i^{new,P1}, F_i^{new,P1} < F_i \\ X_i, F_i^{new,P1} \geq F_i \end{cases} \tag{2}$$

where X_i^{new,P_1} is the new status for the *ith* solution, $X_{i,j}^{new,P_1}$ is its *jth* dimension, $F_i^{new,P1}$ is its objective function value based on first phase of POA.

2.2. Phase 2: Winging On the Surface of the Water (Exploitation Phase)

In this stage, pelicans initiate the process of spreading their wings on the water's surface to encourage their prey to surface. This behavior enables pelicans to capture a greater quantity of fish in the targeted area. The mathematical model describing the updated status of the *ith* pelican candidate solution during phase 2 is presented as follows:

$$x_{(i,j)}^{new,p2} = x_{i,j} + R \times (1 - t/MaxT)(2 \times r - 1) \times x_{i,j} \tag{3}$$

where *t* represents the iteration number, and the total iteration numbers is denoted as $MaxT$. R is a constant value set to 0.2.

In this step, the updates are implemented to either accept or reject the new pelican position, as described by Eq. (4).

$$X_i = \begin{cases} X_i^{new,P2}, F_i^{new,P2} < F_i \\ X_i, F_i^{new,P2} \geq F_i \end{cases} \tag{4}$$

in this context, $X_i^{new,P2}$ represents the updated state of the *ith* obtained solution, $x_{i,j}^{new,p2}$ pertains to its *jth* dimension, and $F_i^{new,P2}$ signifies its objective function value specifically in relation to the second phase of POA.

3. The Proposed Algorithm

In the following subsections, we introduce a novel IPOA algorithm designed to enhance the performance of the original POA algorithm.

3.1 The Levy Flight Strategy

Like many other swarm optimizers, POA can encounter stagnation in sub-optimal regions and exhibit slow convergence, particularly when dealing with intricate and high-dimensional problems. To address these challenges and enhance its search capabilities, two strategies have been adopted, the first one is incorporating Levy flights into the exploration step, and the second one is proposing a novel exponential parameter for a smooth switch between exploration and exploitation.

The proposed IPOA can offer several benefits when dealing with high-dimensional benchmark problems. It is important to point out that Levy flights are a type of random walk with heavy-tailed step lengths that can enhance exploration capabilities (Li et al., 2022). There are two main reasons for using Levy flight:

Improved Exploration: Levy flights introduce long jumps or steps in the search space, which can help the algorithm explore a broader region of the solution space quickly. This is particularly beneficial in high-dimensional spaces where the standard POA may struggle to adequately explore the vast solution space.

Enhanced Global Search: High-dimensional spaces often contain many local optima, making it challenging to find the global optimum. Levy flights can facilitate global search by allowing the algorithm to escape local optima more effectively and discover promising regions of the search space. Consequently, the second part of Eq. (1) has been updated as follows:

$$x_{(i,j)}^{new,p1} = x_{i,j} + V \times \left(x_{i,j} - P_{i,j}\right), \qquad F_{pi} \geq F_i \tag{5}$$

where, V represents the Levy Flight distribution function, and it is computed utilizing Eq. (6):

$$V = \mathcal{g} \times \frac{u \times \sigma}{|v|^{1/\beta}} \tag{6}$$

where $\mathcal{g}$ corresponds to the step size, which is set at a constant value of 0.02. The parameter β is an index and has been specifically assigned a value of 1.5. u and v denote random numbers within the range of 0 to 1. The calculation of σ is determined using Eq. (7):

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sine\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \tag{7}$$

3.2 The Exponential Parameter

After that, the exponential parameter is proposed in this work aiming to improve the search quality of IPOA, ensuring a seamless transition from exploration to exploitation, thereby further improving global movement. Consequently, Equation (3) is updated as follows:

$$x_{(i,j)}^{new,p2} = x_{i,j} + R \times E \times (2 \times r - 1) \times x_{i,j} \tag{8}$$

where E represents the proposed exponential parameter. E can be particularly beneficial in handling multimodal optimization problems, where there are multiple optima. The adaptive nature of E allows the algorithm to explore and exploit multiple modes effectively. E can be calculated using Eq. (9) as follows:

$$E = S \times \exp\left(-C * \left(\frac{t}{MaxT}\right)/Q\right) \tag{9}$$

where S scales the magnitude of the exponential function and set to 0.3, C controls the rate of decay of the exponential function and set to 0.2. Q found in the denominator, affecting how fast the exponential function approaches zero as t increases and is set to 10. Figure 1 illustrate the proposed exponential parameter. One possible adaptive mechanism exhibited by E is its highly oscillatory behavior, as shown in Fig. 1. Because of this oscillation, the algorithm can dynamically switch between exploration and exploitation. Such oscillations guarantee comprehensive search space coverage and keep the algorithm from becoming stuck in a single mode, both of which are advantageous in multimodal optimization. Furthermore, the flowchart detailing the IPOA algorithm is illustrated in Fig. 2.

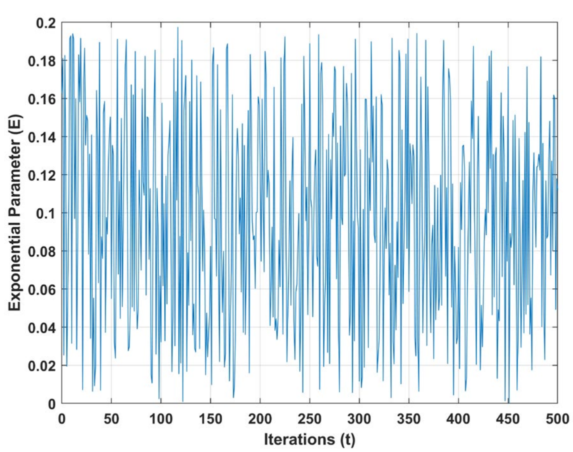


Fig. 1. The proposed exponential parameter (E)

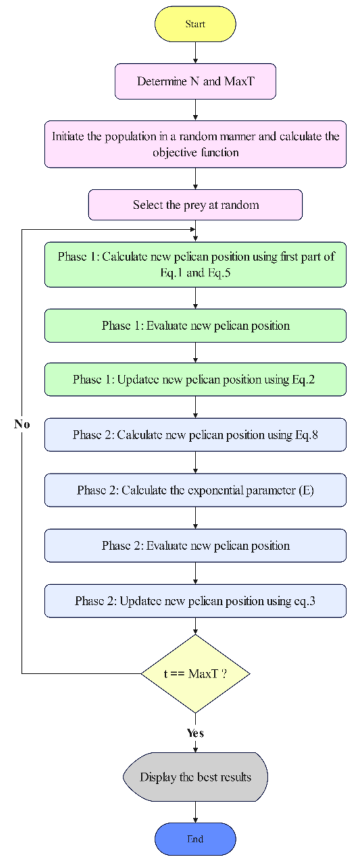


Fig. 2. The flowchart of the proposed IPOA

3.3. Computational Complexity

For the Improved Pelican Optimization Algorithm (IPOA), the time complexity depends on the number of agents N, the number of iterations T, and the computational cost of evaluating the fitness function $F(x)$. During initialization, the algorithm generates N agents, each with d dimensions. This requires $O(N \cdot d)$ operations to randomly initialize the population. Each agent's fitness is evaluated at every iteration. The complexity of a single fitness evaluation is $O(F(x))$, and this is performed N times per iteration for T iterations, leading to a total complexity of $O(T \cdot N \cdot F(x))$. In the exploration phase, a Lévy flight-based update is applied to the agents. The Lévy flight computation has a complexity of $O(d)$ per agent, resulting in a total complexity of $O(T \cdot N \cdot d)$. Similarly, the exploitation phase involves updating agent positions based on the exponential parameter. This also has a complexity of $O(d)$ per agent, contributing another $O(T \cdot N \cdot d)$. Combining all components, the total time complexity of IPOA can be expressed as: $O(T \cdot N \cdot F(x) + O(N \cdot d) + O(T \cdot N))$.

4. Experiments and Results

This section is dedicated to assessing the effectiveness of IPOA in tackling optimization problems and deriving solutions for these challenges. To this end, IPOA is applied to twenty-three standard benchmark functions encompassing various types, including unimodal, high-dimensional multimodal, fixed-dimensional multimodal. Information about the utilized benchmark functions can be founded in (Zeidabadi et al., 2022). To gauge the quality of the results obtained through IPOA, we conduct a comparative analysis against seven well-established algorithms: Whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), Multi-Verse Optimizer (MVO) (Mirjalili et al., 2016), Marine Predators Algorithm (MPA) (Faramarzi et al., 2020), Tunicate Swarm Algorithm (TSA) (Kaur et al., 2020), Harris Hawks Optimizer (HHO) (Heidari et al., 2019), White Shark Optimizer (WSO) (Braik et al., 2022), and the standard POA. In each case, IPOA and the competitor algorithms undergo thirty independent runs, with each run comprising 500 iterations and a fixed population size of 30 for optimizing the objective functions. The optimization results for these objective functions are presented using key statistical indices, including best, worst, mean, and standard deviation (SD). For reference, the control parameter values for the competitor algorithms are listed in Table 1.

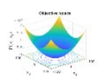
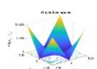
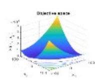
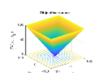
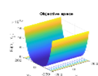
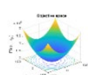
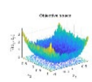
Table 1
Parameter settings

Algorithm	Parameter Values
IPOA	*I*= 1 or 2, R=0.2
POA	*I*= 1 or 2, R=0.2
WOA	*R*= [0,1], *I*= [-1,1], *a*: 2 to 0
MVO	Min (WEP) = 0.2, Max (WEP) = 1, Exploitation accuracy=6
MPA	Fish aggregating devices=0.2, Binary vector=0 or 1, Constant number=0.5
TSA	c_1,c_2,$c_3$in the range [0-1], $P_{min} = 1, P_{max} = 4$
HHO	$E_0 = [-1,1], E: 2\ to\ 0$
WSA	$a_0 = 6.25,\ a_1 = 100,\ a_2 = 0.0005$

4.1. Evaluation of Unimodal Benchmark Functions

The outcomes of applying IPOA and seven competitor algorithms to the unimodal functions F1 to F7 are documented in Table 2.

Table 2
Results of evaluating unimodal functions

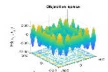
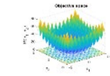
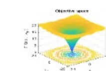
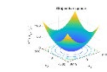
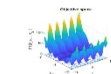
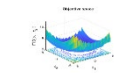
E.T	F.	Figure	In.	IPOA	POA	WOA	MVO	MPA	TSA	HHO	WSO
Unimodal Functions	1		**B**	**2.15E-**	5.87E-	1.07E-84	0.069132	5.22E-23	2.23e−51	1.82E-113	11.3561
			W	3.46E-	8.54E-99	1.53E-74	0.312412	2.28E-20	4.31e−44	3.69E-101	423.8192
			M	**1.21E-**	4.57E-	5.80E-79	0.162312	9.60E-22	3.84e−48	4.46E-107	83.5361
			SD	**0**	8.34E-	1.06E-79	0.0296	1.75E-22	7.01E-49	8.15E-108	15.2515
	2		**B**	**2.98E-**	6.48E-55	3.75E-56	0.17382	1.73E-15	8.48E-39	6.67E-63	0.89156
			W	4.44E-	4.74E-51	1.04E-50	0.51671	2.86E-13	6.81e−26	5.36E-51	6.34651
			M	**2.04E-**	2.05E-53	1.08E-53	0.28527	9.32E-14	2.45e−31	6.15E-55	1.87641
			SD	**3.73E-**	3.74E-54	1.98E-54	0.0521	1.70E-14	4.47E-32	1.12E-55	0.3426
	3		**B**	**1.55E-**	1.68E-	21818.42	4.734351	9.81E-06	1.15E-21	1.19E-94	633.4576
			W	1.61E-	5.09E-	54291.2	27.21344	0.00111	2.91e−11	2.25E-80	3887.45
			M	**1.29E-**	7.50E-	24162.05	13.55651	3.04E-03	4.43e−15	7.06E-84	1617.54
			SD	**0**	1.37E-	4.41E+03	2.4751	5.54E-04	8.09E-16	1.29E-84	295.321
	4		**B**	**2.81E-**	1.56E-56	0.14113	0.167123	1.88E-09	4.17e−5	5.77E-57	12.5412
			W	3.61E-	1.98E-51	76.957	0.914531	1.21E-07	0.035113	1.39E-50	25.4523
			M	**2.92E-**	3.74E-53	40.8523	0.610921	4.09E-08	0.004871	5.22E-54	18.8891
			SD	**5.32E-**	6.84E-54	7.4586	0.1115	7.47E-09	8.89E-04	9.53E-55	3.4487
	5		**B**	**2.12E-05**	26.1878	27.5301	25.4551	25.107	26.53351	0.003425	67.432
			W	24.5868	28.9393	28.765	2232.671	26.0972	28.51838	0.065786	3102.323
			M	**8.20E-01**	28.2561	27.9197	386.349	25.404	27.5176	0.047074	2356.43
			SD	**0.1497**	5.1588	5.0974	70.5374	4.6381	5.024	0.0086	430.223
	6		**B**	**0**	**0**	0.1342	0.081766	3.53E-08	7.15E-23	1.83E-06	13.1242
			W	**0**	**0**	0.8821	0.34156	0.00181	0.429712	0.0005481	733.5619
			M	**0**	**0**	0.3191	0.19561	3.90E-03	4.33E-09	0.0001797	134.761
			SD	**0**	**0**	0.0583	0.0357	7.12E-04	7.90E-10	3.28E-05	24.603
	7		**B**	2.58E-05	6.92E-05	0.0004	0.006123	0.00096	2.73E-05	2.84E-05	**8.67e−07**
			W	1.65E-04	5.80E-01	0.0123	0.030713	0.00224	0.010815	4.09E-02	1.82E-03
			M	8.84E-05	2.71E-03	0.0033	0.021863	0.0018	2.12E-04	1.10E-03	**8.15e−05**
			SD	1.61E-05	4.95E-04	6.02E-04	0.004	3.29E-04	3.88E-05	2.01E-04	**1.49E-05**

An examination of the statistical metrics reveals that IPOA, with its enhanced search capabilities, successfully attains the global optimum for optimizing functions F1 through F6. Additionally, IPOA achieves an optimal solution for F6 similar to that of POA. However, for F7, WSO outperforms other algorithms in obtaining a superior solution. A comprehensive analysis of the simulation outcomes indicates that IPOA excels in optimizing unimodal functions, consistently delivering highly competitive results compared to the other algorithms.

4.2. Evaluation of High-Dimensional Multimodal Benchmark Functions

Table 3 displays the optimization outcomes for high-dimensional multimodal functions F8 to F13, achieved through the application of IPOA and competitor algorithms. The simulation outcomes indicate that IPOA successfully discovers the global optima for functions F9 and F11. Additionally, IPOA demonstrates its superior optimization capabilities by outperforming other algorithms in optimizing functions F8, F10, and F13. The exception is function F12, for which MPA attains superior results. A comparative analysis of competitor algorithms against IPOA underscores IPOA's remarkable efficiency in optimizing multimodal functions, primarily owing to its advanced capabilities.

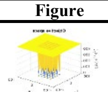
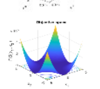
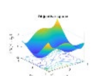
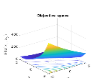
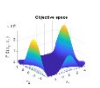
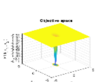
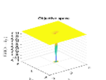
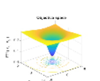
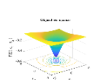
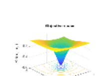
Table 3
Optimization results of high-dimensional multimodal functions

E.T	F.	Figure	In.	IPOA	POA	WOA	MVO	MPA	TSA	HHO	WSO
			B	**-12569.45**	-8118.39	-12203.21	-9275.27	-8772.47	-6685.34	-12569.19	-8912.11
	8		W	-9144.25	-6977.02	-8499.91	-6356.16	-3594.163	-4856.31	-12567.36	-5934.14
			M	-10540.31	-7747.46	-9016.33	-7811.22	-6718.29	-5641.22	**-12568.41**	-7644.15
			SD	370.4686	880.3709	648.708	868.73	1.07E+03	1.26E+03	**0.1899**	899.2326
			B	**0**	**0**	**0**	44.13521	6.43E-145	3.51E-04	**0**	13.322
	9		W	0	0	0	154.7712	6.65E-78	0.0081	0	67.452
			M	**0**	**0**	**0**	103.1234	2.31E-133	9.89E-01	**0**	38.998
			SD	**0**	**0**	**0**	18.8277	4.22E-134	0.1806	**0**	7.12
			B	**8.88E-16**	4.44E-15	**8.88E-16**	0.069781	2.65E-13	2.73E-08	**8.88E-16**	2.1243
Unimodal Functions	10		W	8.88E-16	4.44E-15	7.99E-14	1.879032	4.87E-11	0.0012	8.88E-16	7.0001
			M	**8.88E-16**	4.44E-15	4.44E-15	0.542676	8.45E-12	7.81E-02	**8.88E-16**	4.9343
			SD	**1.62E-16**	8.11E-16	8.11E-16	0.0991	1.54E-12	0.0143	**1.62E-16**	0.9009
			B	**0**	**0**	**0**	0.179045	**0**	4.56E-16	**0**	1.36891
	11		W	0	0	0	0.564978	0	0.0612	0	6.56341
			M	**0**	**0**	**0**	0.435217	**0**	1.12E-07	**0**	1.89672
			SD	**0**	**0**	**0**	0.0795	**0**	2.04E-08	**0**	0.3463
			B	0.0596	0.05847	0.034	0.001315	**3.46E-10**	0.0132	9.28E-06	0.84674
	12		W	0.2923	0.34605	0.059	8.267811	0.0561	17.2313	8.22E-01	563.819
			M	0.1664	0.1894	0.044	1.654517	**2.18E-02**	5.1221	8.92E-02	37.056
			SD	0.0304	0.0346	0.008	0.3021	**0.004**	0.9352	0.0163	6.7655
			B	**4.41E-13**	2.0223	0.1566	0.008113	4.76E-04	2.1375	4.74E-05	11.1298
	13		W	3.45E-07	2.9814	0.9379	0.081711	0.3214	3.7456	0.00671	9613.55
			M	**7.42E-09**	2.4803	0.5695	0.031861	1.22E-01	2.8235	0.000297	5744.34
			SD	**1.35E-09**	0.4528	0.104	0.0058	0.0222	0.5155	5.42E-05	1.05E+03

4.3. Evaluation of Fixed-Dimensional Multimodal Benchmark Functions

Table 4 presents the optimization results obtained using IPOA and seven competitor algorithms when dealing with fixed-dimensional multimodal functions ranging from F14 to F23.

Table 4
Optimization results of fixed high-dimensional multimodal functions

E.T	F.	Figure	In.	IPOA	POA	WOA	MVO	MPA	TSA	HHO	WSO
			B	**0.998**	**0.998**	**0.998**	**0.998004**	**0.998004**	**0.998004**	**0.998**	**0.998004**
	14		W	0.998	1.992	2.982	0.998004	0.998004	10.31927	5.9288	4.286715
			M	**0.998**	1.231	1.442	**0.998004**	**0.998004**	1.89322	3.713	1.35122
			SD	**0**	0.0425	0.0811	**7.30E-07**	**7.30E-07**	0.1634	0.4957	0.0645
	15		B	**0.00030798**	**0.00030749**	0.00045	0.000498	0.000307	0.000308	0.0003248	0.000307
			W	0.00030798	0.0012232	0.00061	0.067821	0.002364	0.057812	0.0004397	0.000317
			M	**0.00030798**	0.00048	0.00051	0.005071	0.000821	0.008824	0.0003789	0.000308
			SD	**0**	3.14E-05	3.69E-05	8.70E-04	9.37E-05	0.0016	1.29E-05	3.65E-09
	16		B	**-1.0316**	**-1.0316**	**-1.0316**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.0316**	**-1.03163**
			W	-1.0316	-1.0316	-1.0316	-1.03163	-1.03163	-1	-1.0316	-1.03163
			M	**-1.0316**	**-1.0316**	**-1.0316**	**-1.03163**	**-1.03163**	-1.12576	**-1.0316**	**-1.03163**
			SD	**0**	**0**	**0**	**0**	**0**	0.0172	**0**	**0**
	17		B	**0.39789**	**0.39789**	**0.39789**	0.397887	0.397887	0.397888	**0.39789**	0.397887
			W	0.39789	0.39789	0.39792	0.397888	0.397887	0.398187	0.39793	0.398048
			M	**0.39789**	**0.39789**	0.3979	0.397887	0.397887	0.39812	0.39791	0.397895
			SD	**0**	**0**	1.83E-06	5.48E-07	5.48E-07	4.20E-05	3.65E-06	9.13E-07
	18		B	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
Unimodal Functions			W	3	3	3	3	3	63.1562	3	3
			M	**3**	**3**	**3**	**3**	**3**	7.3215	**3**	**3**
			SD	**0**	**0**	**0**	**0**	**0**	0.789	**0**	**0**
	19		B	**-3.8628**	**-3.8628**	-3.8614	-3.86278	-3.86278	-3.86278	-3.862	-3.86278
			W	-3.8628	-3.8628	-3.7891	-3.85204	-3.86278	-3.86261	-3.7731	-3.86278
			M	**-3.8628**	**-3.8628**	-3.8228	-3.86058	-3.86278	-3.86262	-3.8573	-3.86278
			SD	**0**	**0**	0.0073	4.05E-04	3.65E-06	3.29E-05	0.001	3.65E-06
	20		B	**-3.22**	-3.322	-3.3212	-3.32199	-3.322	-3.32153	-3.1951	-3.322
			W	-3.22	-3.322	-3.1673	-3.20281	-3.2142	-3.1363	-3.0517	-3.20308
			M	**-3.22**	-3.322	-3.1944	-3.25652	-3.2901	-3.2554	-3.0923	-3.29813
			SD	**0**	0.0186	0.0047	0.0067	0.0128	0.0065	0.0233	0.0143
	21		B	**-10.1532**	-10.1532	-10.1443	-10.1532	**-10.1532**	-10.0743	-5.0385	-10.1532
			W	-10.1532	-5.0552	-5.0549	-2.64519	-5.0331	-2.5435	-5.0548	-3.55781
			M	**-10.1532**	-8.3035	-8.3451	-8.13243	-9.1012	-6.8951	-5.0442	-9.1161
			SD	**0**	0.3377	0.3301	0.3689	0.1921	0.5948	0.9328	0.1893
	22		B	**-10.4029**	-10.4029	-10.394	-10.4029	**-10.4029**	-10.3432	-5.083	-10.4029
			W	-10.4029	-5.0877	-2.7646	-2.8532	-5.1867	-1.8912	-5.0792	-2.5234
			M	**-10.4029**	-8.6941	-5.087	-8.8917	-8.4129	-6.1481	-5.0811	-9.1235
			SD	**0**	0.312	0.9705	0.2759	0.3633	0.7768	0.9716	0.2336
	23		B	**-10.5364**	-10.5363	-10.5322	-10.5364	**-10.5364**	-10.4963	-5.1268	-10.5364
			W	-10.5364	-5.1285	-2.4214	-5.23684	-5.1432	-2.6143	-5.1247	-3.65812
			M	**-10.5364**	-8.1389	-5.1272	-9.21166	-8.6472	-8.1012	-5.1251	-9.67811
			SD	**0**	0.4377	0.9876	0.2419	0.3449	0.4446	0.988	0.1567

These results unambiguously demonstrate IPOA's superiority over all other optimization methods for handling functions F14 to F23. An in-depth performance comparison between IPOA and its competitors reinforces IPOA's remarkable efficiency and exceptional performance in optimizing fixed-dimensional multimodal functions.

4.4. Boxplot Analysis

Boxplot result analysis is a fundamental aspect of data interpretation in various fields such as statistics, and data science. When examining boxplots, the first step is to consider the central tendency of the data. The median, represented by the line within the box, provides an excellent measure of central location. If the median is positioned close to the center of the box, it suggests that the data is symmetrically distributed around this central value. Conversely, if the median is skewed towards one end of the box, it indicates that the data might be skewed or asymmetric.

The boxplots depicting the variations in fitness are presented in Figure 3, providing a visual representation of the outcome distributions produced by each algorithm across thirty runs for the given function. These boxplots offer insights into the data distribution, with a particular focus on the identification of outliers. Several observations can be made from these boxplots, including:

- The body of the box of the IPOA is the shortest one among all algorithms in F1 to F6 and F9, F10, F13 and functions from F14 to F23. The IPOA even has no outlier points in these cases. This demonstrates that the performance of IPOA is very stable;
- Only the IPOA algorithm gets zero values of the standard deviation in F1, F3, F6, F9, F11, and functions from F14 to F23.
- In most of the 23 benchmarks, IPOA provides a stable behavior. On the other hand, most of the other algorithms exhibit instability when optimizing these problems. This can be attributed to the effective exponential parameter design, which allows for dynamic allocation of computational resources to various sub-populations supported by different search strategies (further elaborated in Section 3). With the IPOA algorithm, each run has the capability to adapt to the problem based on acquired information. Consequently, while the search trajectories in each run may vary, IPOA maintains a relatively stable performance.

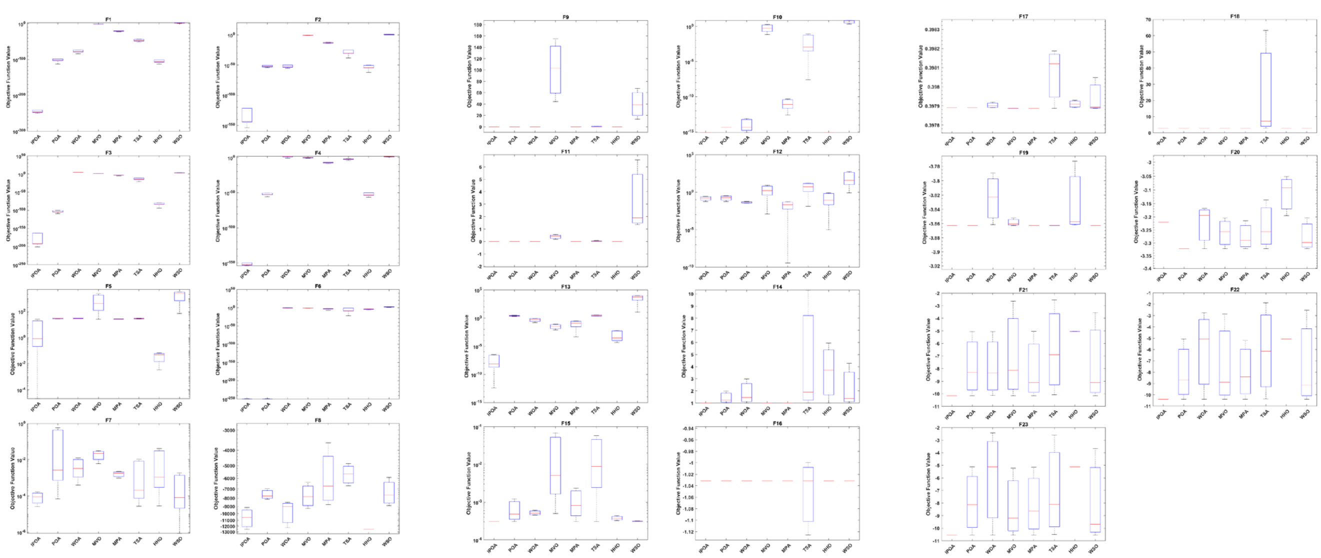


Fig. 3. The boxplot of the objective function variation (F1-F23)

4.5. CEC 2022 Benchmark Functions

In this section, the proposed IPOA approach is evaluated using the CEC2022 benchmark test suite (Luo et al., 2022), alongside the same comparator algorithms used previously. Additionally, the evaluation includes two modified versions of the POA algorithm, namely GPOA (Kusuma & Prasasti, 2022) and CPOA (Song et al., 2023). The research for the CEC2022 benchmark was conducted in accordance with the CEC guidelines. The maximum number of fitness evaluations were 4000, the function's dimensions (Dim) were 10, and each algorithm had been tested 30 times. Table 5 shows the detailed results. "M" and "SD" are the mean and standard deviation of the best-so-far results provided by each algorithm, respectively. The data in Table 5 show that the proposed IPOA algorithm performs optimally on 7 tests.

Table 5
Experimental results of IPOA and 9 comparative algorithms on CEC2022

#	In.	IPOA	POA	WOA	MVO	MPA	TSA	HHO	WSO	GPOA	CPOA
C1	M	3.84E+02	9.95E+02	**3.09E+02**	3.13E+03	2.85E+03	1.83E+03	7.81E+03	4.67E+03	6.43E+03	4.69E+03
	SD	5.31E+01	1.14E+03	**4.81E-01**	2.86E+03	2.36E+04	2.13E+03	2.31E+03	3.51E+03	1.91E+03	3.55E+03
C2	M	**4.13E+02**	4.47E+02	4.44E+02	4.66E+02	1.87E+03	4.67E+02	4.89E+02	4.49E+02	5.81E+02	4.81E+02
	SD	**5.71E+00**	2.56E+01	3.39E+01	4.43E+01	5.89E+02	2.48E+01	2.90E+02	2.89E+01	3.77E+02	2.93E+01
C3	M	**5.98E+02**	6.03E+02	6.22E+02	6.43E+02	6.22E+02	6.24E+02	6.51E+02	6.39E+02	6.59E+02	6.52E+02
	SD	**2.11E-01**	8.42E-01	1.19E+01	6.31E+00	1.31E+01	5.01E+02	2.42E+01	3.71E+00	2.51E+01	3.84E+00
C4	M	**8.22E+02**	8.25E+02	8.38E+02	8.41E+02	8.39E+02	8.48E+02	8.45E+02	8.47E+02	8.49E+02	8.46E+02
	SD	**5.91E+00**	6.47E+00	1.32E+01	6.01E+00	7.88E+00	6.78E+00	1.27E+01	1.26E+01	6.81E+00	1.29E+01
C5	M	**9.05E+02**	1.70E+03	1.23E+03	1.16E+03	9.38E+02	9.77E+02	2.02E+03	9.89E+02	9.83E+02	2.34E+03
	SD	**8.33E+00**	8.43E+01	1.87E+02	1.31E+02	1.67E+01	9.86E+01	3.21E+02	6.43E+01	9.96E+01	3.37E+02
C6	M	**4.48E+03**	6.50E+03	6.71E+03	4.30E+03	5.42E+08	6.52E+03	6.73E+03	4.32E+03	5.45E+08	6.75E+03
	SD	**2.18E+03**	2.20E+03	2.88E+03	2.23E+03	4.81E+08	2.30E+03	2.91E+03	2.28E+03	4.89E+08	2.94E+03
C7	M	**2.05E+03**	2.25E+03	2.26E+03	2.47E+03	2.61E+03	2.44E+03	2.51E+03	2.69E+03	2.52E+03	2.72E+03
	SD	**5.09E+00**	9.93E+00	2.59E+01	3.43E+01	2.23E+01	1.86E+01	3.72E+01	2.34E+01	3.73E+01	2.51E+01
C8	M	**2.20E+03**	2.67E+03	2.45E+03	2.55E+03	2.71E+03	2.73E+03	2.74E+03	2.75E+03	2.70E+03	2.69E+03
	SD	**3.81E+00**	6.52E+00	2.01E+01	5.17E+00	6.32E+00	6.81E+00	3.11E+01	4.24E+00	5.31E+00	3.21E+01

4.6. Wilcoxon Rank Sum Test Analysis

The Wilcoxon rank sum test, also known as the Mann-Whitney U test, is a statistical method used to compare two independent groups or samples to determine if there is a significant difference between them. It is a non-parametric test, meaning that it does not assume a normal distribution of data. In the context of the analysis presented in this study, the Wilcoxon Rank Sum Test is employed to assess and compare the performance of the proposed IPOA algorithm with several competitor algorithms across various optimization functions. The results of the Wilcoxon rank sum test are typically presented in tabular form, as shown in Tables 6 and 7. These tables provide information on the statistical significance of the performance differences between algorithms. The p-values obtained from the test are compared to a predefined significance level (alpha), often set at 0.05. If the p-value is less than alpha, it indicates that there is a significant difference between the two groups being compared. In the context of this study, when all p-values are less than 0.05, it suggests that IPOA significantly outperforms all the competitor algorithms across all optimization functions. This statistical analysis adds a rigorous and objective dimension to the performance evaluation of the proposed IPOA.

Table 6
p-values obtained from Wilcoxon rank sum test for tables 2, 3, and 4

Functions Type	Compared Algorithms						
	IPOA vs POA	**IPOA vs WOA**	**IPOA vs MVO**	**IPOA vs MPA**	**IPOA vs TSA**	**IPOA vs HHO**	**IPOA vs WSO**
Unimodal	3.1250E-02	4.4350E-02	1.5625E-03	2.2341E-02	4.1347E-02	4.0243E-02	3.1553E-02
High dimensional multimodal	3.7500E-02	4.153E-02	3.1251E-02	4.5750E-02	3.1238E-02	5.000E-01	3.6796E-03
Fixed dimensional multimodal	3.5534E-02	5.4977E-03	4.6228E-02	4.6229E-02	3.0758E-01	1.7046E-03	3.3535E-02

Table 7
p-values obtained from Wilcoxon rank sum test for table 5

Functions Type	Compared Algorithms								
	IPOA vs POA	**IPOA vs WOA**	**IPOA vs MVO**	**IPOA vs MPA**	**IPOA vs TSA**	**IPOA vs HHO**	**IPOA vs WSO**	**IPOA vs GPOA**	**IPOA vs CPOA**
CEC2022	3.3507E-02	4.5147E-02	2.2431E-03	3.1352E-02	3.2123E-02	4.6523E-02	3.1924E-02	4.0341E-02	4.1522E-02

5. Application of IPOA for Engineering Design Problems

In this section, the IPOA algorithm's performance is put to the test in real-world engineering design problems encompassing tension/compression spring design, tubular column design, speed reducer design, welded beam design, and I-shaped beam design.

When dealing with constrained optimization problems, if the decision variable values exceed the permissible range, they are adjusted to the boundary values of the intervals. Additionally, to meet both unequal and equal constraints, a penalty coefficient is utilized and incorporated into the objective function value. Consequently, solutions that do not conform to the specified constraints are considered unsuitable solutions (Sadeeq & Abdulazeez, 2023a).

5.1. Tension/ Compression Spring Design Optimization Problem (TCSD)

TCSD problem presents a significant challenge in engineering design, as it involves the optimization of critical components used in various mechanical systems. The goal is to find the optimal design parameters for a tension/compression spring, such as wire diameter and number of active coils, that satisfy specific performance criteria while minimizing the spring's weight. TCSD is of paramount importance in engineering applications where the performance and efficiency of springs play a crucial role, such as in automotive suspensions, industrial machinery, and aerospace systems (Wang et al., 2023). The illustration of this problem's setup is depicted in Figure 4A. The outcomes achieved from the IPOA algorithm and its competition on the optimization of this particular challenge are displayed in Tables 8 and 9. Upon analyzing the simulation results, it becomes evident that IPOA has successfully provided the optimal solution for this design, with the recommended values for the design variables being (0.0518488, 0.360503, 11.0778) resulting in an objective function value of 0. 0.012601. Statistical analysis indicates that IPOA outperforms the competing algorithms when it comes to addressing the TCSD problem. The algorithms' performance rankings are determined by the mean index. The convergence behavior of IPOA and other algorithms while tackling this problem is graphically represented in Figure 5A.

Table 8
Performance of IPOA and competitors' algorithms for the TCSD

Algorithm	Optimal Variables			Optimal Cost
	d	D	P	
IPOA	0.0518488	0.360503	11.0778	0.012601
POA	0.052923	0.38713	9.7062	0.012693
WOA	0.05311	0.33567	15.012	0.013197
MVO	0.052716	0.3819	9.9532	0.012686
MPA	0.054367	0.42403	8.2271	0.012818
TSA	0.056278	0.47682	6.6424	0.013052
HHO	0.054098	0.41738	8.4661	0.012784
WSO	0.06064	0.61262	4.2424	0.014062

Table 9
Statistical results of IPOA and competitors' algorithms for the TCSD

Algorithm	Best	Worst	Mean	Std	Rank
IPOA	0.012601	0.012676	0.012666	1.1867322e-05	1
POA	0.012693	0.013002	0.012832	4.2174636e-05	2
WOA	0.013197	0.017764	0.014713	3.8559668e-04	6
MVO	0.012686	0.018374	0.016761	7.5950861e-04	8
MPA	0.012818	0.013001	0.012993	7.1569080e-05	4
TSA	0.013052	0.013668	0.013321	1.3145341e-04	5
HHO	0.012784	0.012997	0.012842	4.4000378e-05	3
WSO	0.014062	0.016463	0.015152	4.6574674e-04	7

5.2. Tubular Column Design Optimization Problem (TCD)

TCD is a classic engineering challenge that aims to find the optimal dimensions and specifications for a tubular column, typically used in structural engineering applications. The primary objective is to minimize the cost of the column while satisfying various engineering constraints related to strength, stability, and material properties. Th is problem has two design variables, the mean diameter of the column $d(=x1)$ and the thickness of tube $t(= x2)$ (Gao et al., 2020), which are shown in Fig. 4B. The outcomes of applying IPOA and rival algorithms to the TCD problem are displayed in Tables 10 and 11. According to the simulation outcomes, IPOA has achieved the best solution alongside with MPA, with specific values for the design variables at 5.4522 and 0.29163, resulting in a corresponding objective function value of 26.486361. An analysis of the statistical results clearly indicates that IPOA with MPA outperforms the competitor algorithms in resolving the TCD. Figure 5B illustrates the convergence curve of IPOA while obtaining the solution for this design, demonstrating its efficiency.

Table 10

Performance of IPOA and competitors' algorithms for the TCD

Algorithm	Optimal Variables		Optimal Cost
	d	t	
IPOA	5.4522	0.29163	26.486361
POA	5.4521	0.29163	26.4864
WOA	5.45548	0.291685	26.48856
MVO	5. 45138	0.291973	26.53413
MPA	5.45218	0.29162	26.486361
TSA	5.4477	0.29234	26.5028
HHO	5.4504	0.29191	26.4929
WSO	5.44247	0.29317	26.5218

Table 11

Statistical results of IPOA and competitors' algorithms for the TCD

Algorithm	Best	Worst	Mean	Std	Rank
IPOA	26.486361	26.486361	26.486361	0	1
POA	26.4864	28.18624	27.31221	0.1507787087	7
WOA	26.48856	27.4624	26.88077	0.0720089020	6
MVO	26.53413	26.56583	26.54376	0.0104795756	3
MPA	26.486361	26.486361	26.486361	0	1
TSA	26.5028	26.5134	26.5237	0.0068171375	2
HHO	26.4929	26.6075	26.5527	0.0121117889	4
WSO	26.5218	29.14386	26.8492	0.0662450351	5

5.3. Speed Reducer Design Optimization Problem (SRD)

SRD is a real-world engineering challenge, involves the task of optimizing the design parameters of a speed reducer system to meet specific performance criteria. In this problem, the design variables typically include geometric parameters like gear tooth profiles, diameters, and other dimensions of the speed reducer components. The objective is to minimize the weight of the speed reducer (Jiang et al., 2022). The optimization results for the SRD, obtained using the IPOA algorithm and compared with those of competing algorithms, are summarized in Tables 12 and 13. Based on the simulation outcomes, it is evident that IPOA has achieved the optimal solution for this design. The suggested values for the design variables, which yield an optimal performance, include (3.46124, 0.7, 17, 7.35871, 7.71735, 3.35867, 5.28892), resulting in an objective function value of 2992.312. An analysis of the statistical results reveals that the IPOA approach has consistently outperformed the competitor algorithms in terms of various statistical indicators. These findings emphasize the superior performance of IPOA in addressing the SRD Problem. The problem's schematic is presented in Figure 4C, while the convergence curve of IPOA during the optimization process for this design is depicted in Figure 5C.

Table 12

Performance of IPOA and competitors' algorithms for the SRD

Algorithm	Optimal Variables							Optimal Cost
	b	m	p	l_1	l_2	d_1	d_2	
IPOA	3.46124	0.7	17	7.35871	7.71735	3.35867	5.28892	2992.312
POA	3.48068	0.7	17	7.30018	7.72772	3.34879	5.28712	2996.996
WOA	3.36652	0.7	17	7.30008	7.7108	3.38374	5.28665	3008.651
MVO	3.39909	0.700005	17.0008	8.12903	7.45968	3.3522	5.28662	3011.341
MPA	3.48048	0.700065	17.002	7.30732	7.95997	3.36104	5.28674	3002.113
TSA	3.36426	0.7	17.0003	7.65956	7.85765	3.35174	5.28699	3006.158
HHO	3.42545	0.7	17	7.38004	7.68411	3.3445	5.28761	3003.908
WSO	3.31275	0.7	17	7.71889	7.8618	3.34949	5.28784	3010.781

Table 13
Statistical results of IPOA and competitors' algorithms for the SRD

Algorithm	Best	Worst	Mean	Std	Rank
IPOA	2992.312	2998.981	2996.455	0.75640485191	1
POA	2996.996	3002.761	3000.462	1.48797961455	2
WOA	3008.651	3231.474	3109.378	21.3732296389	8
MVO	3011.341	3081.784	3034.331	7.67158471460	6
MPA	3002.113	3008.341	3006.912	2.66558311319	3
TSA	3006.158	3020.483	3013.312	3.83405790253	4
HHO	3003.908	3224.471	3101.321	19.9022294236	7
WSO	3010.781	3075.439	3029.971	6.87556126436	5

5.4. Welded Beam Design Optimization Problem (WBD)

WBD Problem is a structural engineering design aims to minimize the fabrication cost of the welded beam. The results obtained using the IPOA algorithm, along with comparisons to competing algorithms, are exhaustively presented in Tables 14 and 15. These results clearly demonstrate IPOA's capability in successfully achieving the optimal solution for this intricate design problem. The values of the design variables that correspond to this optimal solution are (0.20434, 3.5002, 9.0391, 0.20572) resulting in a remarkable objective function value of 1.7017. The comprehensive statistical analyses of these results reinforce the conclusion that IPOA consistently outperforms its competitor algorithms. This underscores IPOA's prowess in addressing complex structural optimization problems (Połap & Woźniak, 2021). Fig. 4D provides a schematic representation of the WBD Problem. Additionally, for a visual representation of IPOA's iterative optimization process for this challenge, please refer to Fig. 5D. This convergence curve offers a dynamic view of how IPOA systematically hones in on the optimal solution over successive iterations.

Table 14
Performance of IPOA and competitors' algorithms for the WBD

Algorithm	Optimal Variables				Optimal Cost
	h	l	T	b	
IPOA	0.20434	3.5002	9.0391	0.20572	1.7017
POA	0.20567	3.4718	9.0415	0.20571	1.7256
WOA	0.20486	3.4514	9.1647	0.20599	1.8275
MVO	0.2034	3.5206	9.0383	0.20572	1.7282
MPA	0.20586	3.4691	9.0349	0.20583	1.7255
TSA	0.20119	3.5711	9.0366	0.20573	1.7314
HHO	0.20927	3.4299	8.9476	0.20993	1.7326
WSO	0.20487	3.4946	9.0386	0.20572	1.7251

Table 15
Statistical results of IPOA and competitors' algorithms for the WBD

Algorithm	Best	Worst	Mean	Std	Rank
IPOA	1.7017	1.7257	1.7167	0.002738612	1
POA	1.7256	1.7313	1.7279	0.004783443	4
WOA	1.8275	4.2321	2.4610	0.138628579	8
MVO	1.7282	1.7812	1.7413	0.007229937	5
MPA	1.7255	1.7275	1.7269	0.004600869	2
TSA	1.7314	1.7536	1.7457	0.008033264	7
HHO	1.7326	1.7936	1.7546	0.009658174	6
WSO	1.7251	1.7341	1.7271	0.004637384	3

5.5. I-Shaped Beam Design Optimization Problem (I-SBD)

Another common engineering optimization problem is the I-beam design problem, which has the objective of minimizing the vertical deflection of the beam as depicted in Fig. 4E. This problem also involves satisfying both cross-sectional area and stress constraints under predefined loads. The variables considered in this problem include the width of the flange (denoted as *b* and represented as x1), the height of the section (referred to as *h* and denoted as x2), the thickness of the web (designated as *tw* and expressed as x3), and the thickness of the flange (*tf*, represented as x4) (Chen et al., 2023). The results of applying the IPOA algorithm alongside competing algorithms to address the I-SBD problem are presented in Tables 16 and 17. Based on the simulation findings, IPOA has emerged as the top-performing solution, yielding specific values for the design variables: *b* = 80, *h* = 50, *tw* = 0.900058, and *tf* = 2.32175, resulting in a corresponding objective function value of 0.013074. An examination of the statistical outcomes unambiguously demonstrates IPOA's superiority over rival algorithms in tackling the I-beam design problem. Fig. 5E visually portrays IPOA's convergence curve during the process of obtaining the solution for this design, underscoring its effectiveness.

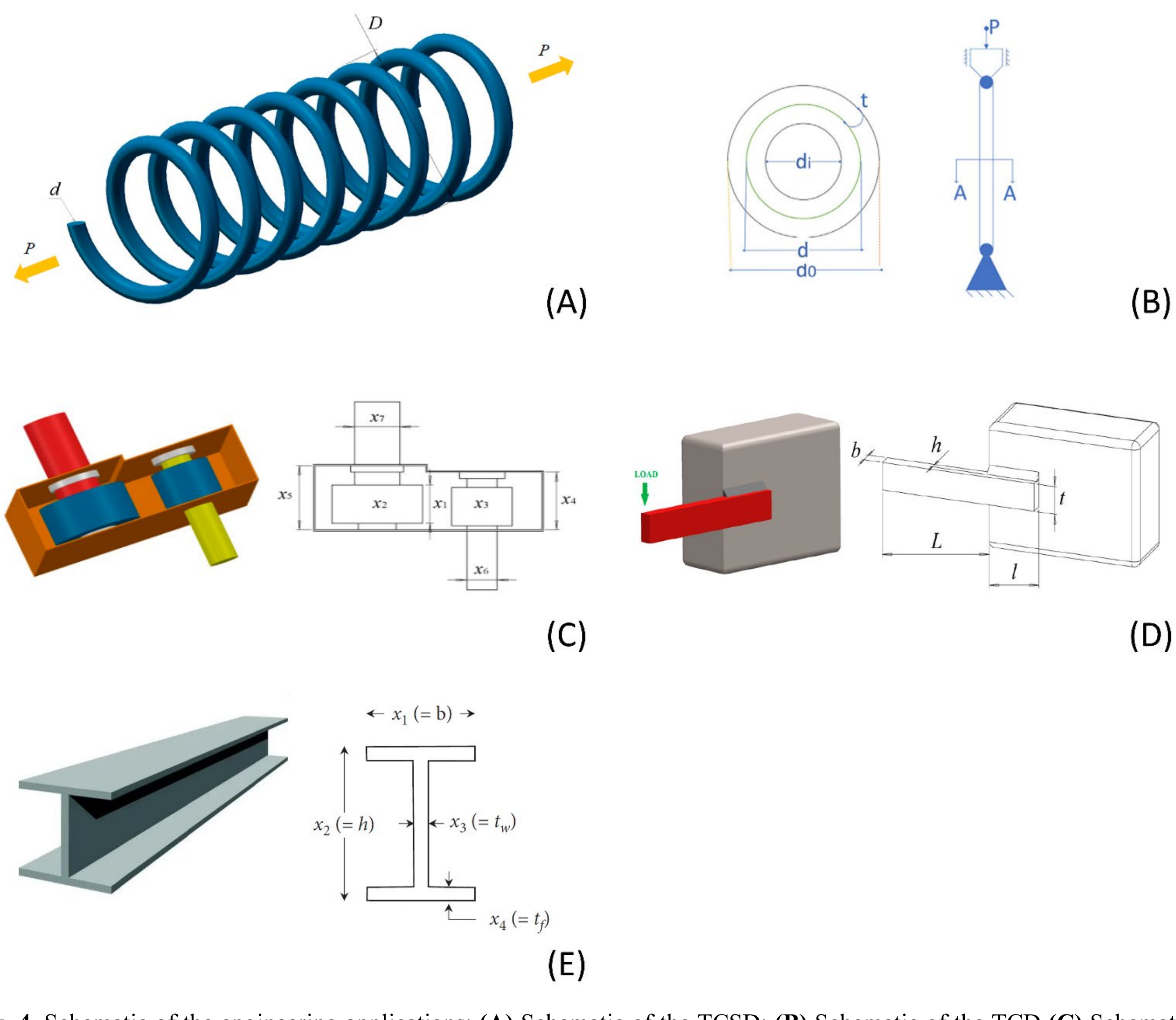


Fig. 4. Schematic of the engineering applications: **(A)** Schematic of the TCSD; **(B)** Schematic of the TCD **(C)** Schematic of the SRD; **(D)** Schematic of the WBD; **(E)** Schematic of the I-SBD.

Table 16
Performance of IPOA and competitors' algorithms for the I-SBD

Algorithm	Optimal Variables				Optimal Cost
	x_1	x_2	x_3	x_4	
IPOA	80	50	0.900058	2.32175	0.013074
POA	80	50	0.9	2.32179	0.013074
WOA	80	50	0.90006	2.31638	0.013088
MVO	80	48.4151	0.900017	2.399266	0.013076
MPA	80	49.999	0.9	2.32179	0.013074
TSA	79.9875	49.9922	0.91872	2.30694	0.013112
HHO	80	50	0.900049	2.32175	0.013074
WSO	79.8950	49.9344	1.151839	2.12357	0.013074

Table 17
Statistical results of IPOA and competitors' algorithms for the I-SBD

Algorithm	Best	Worst	Mean	Std	Rank
IPOA	0.013074	0.013074	0.013074	0	1
POA	0.013074	0.013343	0.013112	6.93781906e-06	3
WOA	0.013088	0.083128	0.016171	5.65432253e-04	6
MVO	0.013076	0.069102	0.015842	5.05365346e-04	5
MPA	0.013074	0.013076	0.013075	1.82574185e-07	2
TSA	0.013112	0.149339	0.024921	0.002162956	8
HHO	0.013074	0.013423	0.013318	4.45481013e-05	4
WSO	0.013074	0.060413	0.016419	6.10710651e-04	7

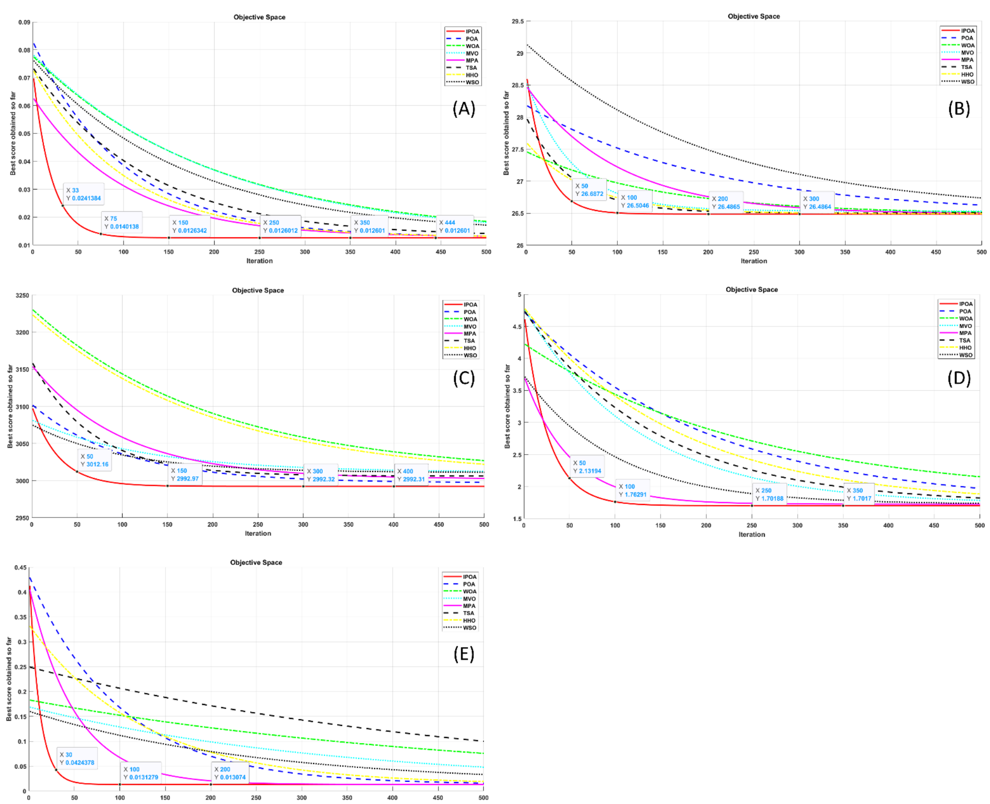


Fig. 5. Convergence rate diagram for IPOA and other competing algorithms: **(A)** Convergence rate diagram for TCSD; **(B)** Convergence rate diagram for TCD **(C)** Convergence rate diagram for SRD; **(D)** Convergence rate diagram for WBD; **(E)** Convergence rate diagram for I-SBD.

6. Conclusions

This paper introduces a novel variant of the Pelican Optimization Algorithm (POA) termed IPOA, aimed at addressing a wide range of optimization challenges. To enhance the exploration phase, IPOA incorporates a novel transition mechanism. Additionally, it employs a new random position strategy to bolster local search capabilities, thereby preventing the algorithm from becoming trapped in local optima. These enhancements collectively promote a more balanced exploration-exploitation trade-off within the proposed algorithm. To comprehensively assess its function optimization performance, IPOA is benchmarked against the original POA and six advanced metaheuristics using 23 classical benchmark functions. Also, CEC2022 benchmark have been used to further prove the performance of the proposed IPOA. Statistical significance is confirmed through the Wilcoxon rank-sum test. Numerical and statistical results affirm that IPOA significantly outperforms other algorithms in terms of accuracy, convergence speed, stability, and avoidance of local optima. To showcase its practical applicability, IPOA is applied to five engineering design problems, yielding highly competitive solutions. As future perspectives, the research envisions the development of multi-objective and binary versions of IPOA. Furthermore, the algorithm's potential applications span diverse fields, including network applications, text clustering, parameter estimation, feature selection, and more. While the results showcase IPOA's excellent performance, its reliance on fixed parameter settings limits its adaptability to different problem landscapes. As a part of future research, we aim to focus on developing parameter-adaptive adjustment strategies that allow IPOA to dynamically tune its parameters based on the optimization context. This enhancement is expected to further improve the algorithm's robustness and adaptability, enabling it to tackle a broader range of dynamic and real-world optimization challenges.

The statement of data

The corresponding author will provide the data sets upon request.

Conflicts of interest

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

References

Abu-Hashem, M., & Shambour, M. (2024). An improved black widow optimization (IBWO) algorithm for solving global optimization problems. *International Journal of Industrial Engineering Computations*, *15*(3), 705–720.

Ajagekar, A., Al Hamoud, K., & You, F. (2022). Hybrid Classical-Quantum Optimization Techniques for Solving Mixed-Integer Programming Problems in Production Scheduling. *IEEE Transactions on Quantum Engineering*, *3*(March), 1–16. https://doi.org/10.1109/TQE.2022.3187367

Al-Betar, M. A., Awadallah, M. A., Braik, M. S., Makhadmeh, S., & Doush, I. A. (2024). Elk herd optimizer: a novel nature-inspired metaheuristic algorithm. In *Artificial Intelligence Review* (Vol. 57, Issue 3). Springer Netherlands. https://doi.org/10.1007/s10462-023-10680-4

Alamir, N., Kamel, S., Megahed, T. F., Hori, M., & Abdelkader, S. M. (2023). Developing Hybrid Demand Response Technique for Energy Management in Microgrid Based on Pelican Optimization Algorithm. *Electric Power Systems Research*, *214*(PA), 108905. https://doi.org/10.1016/j.epsr.2022.108905

Alghamdi, A. S. (2024). Cost-Effective Planning of Hybrid Energy Systems Using Improved Horse Herd Optimizer and Cloud Theory under Uncertainty. In *Electronics* (Vol. 13, Issue 13). https://doi.org/10.3390/electronics13132471

Amine Tahiri, M., Zohra El hlouli, F., Bencherqui, A., Karmouni, H., Amakdouf, H., Sayyouri, M., & Qjidaa, H. (2023). White blood cell automatic classification using deep learning and optimized quaternion hybrid moments. *Biomedical Signal Processing and Control*, *86*(PA), 105128. https://doi.org/10.1016/j.bspc.2023.105128

Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, *243*, 108457. https://doi.org/10.1016/j.knosys.2022.108457

Chen, L., Zhao, B., & Ma, Y. (2023). FSSSA: A Fuzzy Squirrel Search Algorithm Based on Wide-Area Search for Numerical and Engineering Optimization Problems. *Mathematics*, *11*(17), 3722. https://doi.org/10.3390/math11173722

Dao, T.-K., Ngo, T.-G., Pan, J.-S., Nguyen, T.-T.-T., & Nguyen, T.-T. (2024). Enhancing Path Planning Capabilities of Automated Guided Vehicles in Dynamic Environments: Multi-Objective PSO and Dynamic-Window Approach. *Biomimetics*, *9*(1), 35.

Emam, M. M., Houssein, E. H., & Ghoniem, R. M. (2023). A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images. *Computers in Biology and Medicine*, *152*(October 2022), 106404. https://doi.org/10.1016/j.compbiomed.2022.106404

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, *152*, 113377. https://doi.org/10.1016/j.eswa.2020.113377

Gandomi, A. H., & Deb, K. (2020). Implicit constraints handling for efficient search of feasible solutions. *Computer Methods in Applied Mechanics and Engineering*, *363*, 112917. https://doi.org/https://doi.org/10.1016/j.cma.2020.112917

Gao, C., Hu, Z., Xiong, Z., & Su, Q. (2020). Grey prediction evolution algorithm based on accelerated even grey model. *IEEE Access*, *8*, 107941–107957.

Hashish, M. S., Hasanien, H. M., Ullah, Z., Alkuhayli, A., & Badr, A. O. (2023). Giant Trevally Optimization Approach for Probabilistic Optimal Power Flow of Power Systems Including Renewable Energy Systems Uncertainty. *Sustainability*, *15*(18), 13283.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872.

Houssein, E. H., Oliva, D., Samee, N. A., Mahmoud, N. F., & Emam, M. M. (2023). Liver Cancer Algorithm: A novel bio-inspired optimizer. *Computers in Biology and Medicine*, 107389.

Jiang, Y., Wu, Q., Zhu, S., & Zhang, L. (2022). Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems. *Expert Systems with Applications*, *188*(April 2021), 116026. https://doi.org/10.1016/j.eswa.2021.116026

Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, *90*(December 2018), 103541. https://doi.org/10.1016/j.engappai.2020.103541

Kuang, X., Hou, J., Liu, X., Lin, C., Wang, Z., & Wang, T. (2024). Improved African Vulture Optimization Algorithm Based on Random Opposition-Based Learning Strategy. In *Electronics* (Vol. 13, Issue 16). https://doi.org/10.3390/electronics13163329

Kusuma, P. D., & Prasasti, A. L. (2022). Guided Pelican Algorithm. *International Journal of Intelligent Engineering and Systems*, *15*(6), 179–190. https://doi.org/10.22266/ijies2022.1231.18

Latifi Amoghin, M., Abbaspour-Gilandeh, Y., Tahmasebi, M., Kaveh, M., El-Mesery, H. S., Szymanek, M., & Sprawka, M. (2024). VIS/NIR Spectroscopy as a Non-Destructive Method for Evaluation of Quality Parameters of Three Bell Pepper Varieties Based on Soft Computing Methods. In *Applied Sciences* (Vol. 14, Issue 23). https://doi.org/10.3390/app142310855

Le Digabel, S., & Wild, S. M. (2023). A taxonomy of constraints in black-box simulation-based optimization. *Optimization and Engineering*. https://doi.org/10.1007/s11081-023-09839-3

Li, J., An, Q., Lei, H., Deng, Q., & Wang, G. G. (2022). Survey of Lévy Flight-Based Metaheuristics for Optimization. *Mathematics*, *10*(15). https://doi.org/10.3390/math10152785

Luo, W., Lin, X., Li, C., Yang, S., & Shi, Y. (2022). Benchmark functions for CEC 2022 competition on seeking multiple optima in dynamic environments. *ArXiv Preprint ArXiv:2201.00523*.

Mataifa, H., Krishnamurthy, S., & Kriger, C. (2022). Volt/VAR Optimization: A Survey of Classical and Heuristic Optimization Methods. *IEEE Access*, *10*, 13379–13399. https://doi.org/10.1109/ACCESS.2022.3146366

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*(2), 495–513. https://doi.org/10.1007/s00521-015-1870-7

Mohammed, G. P., Alasmari, N., Alsolai, H., Alotaibi, S. S., Alotaibi, N., & Mohsen, H. (2022). Autonomous Short-Term Traffic Flow Prediction Using Pelican Optimization with Hybrid Deep Belief Network in Smart Cities. *Applied Sciences (Switzerland)*, *12*(21). https://doi.org/10.3390/app122110828

Parvathi, K. A., Kotaiah, N. C., & Rani, K. R. (2022). Pelican Optimization Algorithm for Optimal Demand Response in Islanded Active Distribution Network Considering Controllable Loads. *International Journal of Intelligent Engineering and Systems*, *15*(6), 132–141. https://doi.org/10.22266/ijies2022.1231.14

Połap, D., & Woźniak, M. (2021). Red fox optimization algorithm. *Expert Systems with Applications*, *166*(October 2020), 114107. https://doi.org/10.1016/j.eswa.2020.114107

Rabie, A. H., Mansour, N. A., & Saleh, A. I. (2023). Leopard seal optimization (LSO): A natural inspired meta-heuristic algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *125*, 107338. https://doi.org/10.1016/j.cnsns.2023.107338

Sadeeq, H. T., & Abdulazeez, A. M. (2022a). Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems. *IEEE Access*, *October*, 121615–121640. https://doi.org/10.1109/ACCESS.2022.3223388

Sadeeq, H. T., & Abdulazeez, A. M. (2022b). *Improved Northern Goshawk Optimization Algorithm for Global Optimization*. 89–94.

Sadeeq, H. T., & Abdulazeez, A. M. (2023a). Car side impact design optimization problem using giant trevally optimizer. *Structures*, *55*(February), 39–45. https://doi.org/10.1016/j.istruc.2023.06.016

Sadeeq, H. T., & Abdulazeez, A. M. (2023b). Metaheuristics: A Review of Algorithms. *International Journal of Online and Biomedical Engineering*, *19*(9), 142–164. https://doi.org/10.3991/ijoe.v19i09.39683

Saleem, S., & Gallagher, M. (2022). Using regression models for characterizing and comparing black box optimization problems. *Swarm and Evolutionary Computation*, *68*(June 2021), 100981. https://doi.org/10.1016/j.swevo.2021.100981

Shehadeh, H. A. (2023). Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization. *Neural Computing and Applications*, *35*(15), 10733–10749. https://doi.org/10.1007/s00521-023-08261-1

Song, H.-M., Xing, C., Wang, J.-S., Wang, Y.-C., Liu, Y., Zhu, J.-H., & Hou, J.-N. (2023). Improved pelican optimization algorithm with chaotic interference factor and elementary mathematical function. *Soft Computing*, *27*(15), 10607–10646. https://doi.org/10.1007/s00500-023-08205-w

Tian, T., Liang, Z., Wei, Y., Luo, Q., & Zhou, Y. (2024). Hybrid Whale Optimization with a Firefly Algorithm for Function Optimization and Mobile Robot Path Planning. *Biomimetics*, *9*(1), 39.

Trojovský, P., & Dehghani, M. (2022). Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors*, *22*(3). https://doi.org/10.3390/s22030855

Wan, Y., Zuo, T. Y., Chen, L., Tang, W. C., & Chen, J. (2020). Efficiency-Oriented Production Scheduling Scheme: An Ant Colony System Method. *IEEE Access*, *8*, 19286–19296. https://doi.org/10.1109/ACCESS.2020.2968378

Wang, B., Jin, Q., Zhao, R., & Zhang, Y. (2023). A New Optimization Idea: Parallel Search-based Golden Jackal Algorithm. *IEEE Access*, *11*(August), 1–1. https://doi.org/10.1109/access.2023.3312684

Wang, J., Wang, W. C., Chau, K. W., Qiu, L., Hu, X. X., Zang, H. F., & Xu, D. M. (2024). An Improved Golden Jackal Optimization Algorithm Based on Multi-strategy Mixing for Solving Engineering Optimization Problems. *Journal of Bionic Engineering*, *21*(2), 1092–1115. https://doi.org/10.1007/s42235-023-00469-0

Wolpert, D., & Macready, W. (1997). No Free Lunch Theorems for Optimization. *Evolutionary Computation, IEEE Transactions On*, *1*, 67–82.

Wongvanich, N., Roongmuanpha, N., & Tangsrirat, W. (2023). Extended Exploration Grey Wolf Optimization, CFOA-Based Circuit Implementation of the sigr Function and its Applications in Finite-Time Terminal Sliding Mode Control. *IEEE Access*, *11*, 88388–88402. https://doi.org/10.1109/ACCESS.2023.3305943

Yang, H., Yang, X., & Li, G. (2023). Dual feature extraction system for ship-radiated noise and its application extension. *Ocean Engineering*, *285*(P2), 115352. https://doi.org/10.1016/j.oceaneng.2023.115352

Yu, Y., Yao, M., Huang, J., & Xiao, X. (2024). When Process Analysis Technology Meets Transfer Learning: A Model Transfer Strategy Between Different Spectrometers for Quantitative Analysis. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–19. https://doi.org/10.1109/TIM.2024.3353273

Yuan, X., Karbasforoushha, M. A., Syah, R. B. Y., Khajehzadeh, M., Keawsawasvong, S., & Nehdi, M. L. (2023). An Effective Metaheuristic Approach for Building Energy Optimization Problems. *Buildings*, *13*(1). https://doi.org/10.3390/buildings13010080

Zeidabadi, F. A., Dehghani, M., Trojovský, P., Hubálovský, Š., Leiva, V., & Dhiman, G. (2022). Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems. *Computers, Materials and Continua*, *72*(1), 399–416. https://doi.org/10.32604/cmc.2022.024736

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, 114194. https://doi.org/10.1016/j.cma.2021.114194

Zhong, K., Xiao, F., & Gao, X. (2024). APFA: Ameliorated Pathfinder Algorithm for Engineering Applications. *Journal of Bionic Engineering*, *0123456789*. https://doi.org/10.1007/s42235-024-00510-w

Zhong, M., Wen, J., Ma, J., Cui, H., Zhang, Q., & Parizi, M. K. (2023). A hierarchical multi-leadership sine cosine algorithm to dissolving global optimization and data classification: The COVID-19 case study. *Computers in Biology and Medicine*, *164*(June), 107212. https://doi.org/10.1016/j.compbiomed.2023.107212