Improved Tuna Swarm Optimization (ITSO) Algorithm based on Adaptive Fitness-Weight for Global Optimization

A. F. Seini Yussif[1*], T. Seini[2], C. Adu[3]
[1*]Department of Electrical and Electronics Engineering, University for Development Studies, Tamale, Ghana
[2]Department of Physical Sciences, University for Development Studies, Tamale, Ghana
[3]Department of Computer and Electrical Engineering, University of Energy and Natural Resources, Sunyani, Ghana
[*]corresponding author's email: seini.coe@gmail.com

Abstract – *In this paper, an improved variant of the Tuna Swarm Optimization (TSO) algorithm called the Improved Tuna Swarm Optimization (ITSO) algorithm is proposed. An alternative way to determine the weight (p) value to improve the convergence speed of the original Tuna Swarm Optimization (TSO) algorithm is the primary objective. This method comes along with an adaptive fitness weight strategy which is used to replace the former approach for the weight value calculation to significantly improve the algorithm's performance on high-dimensional problems. The proposed ITSO, the original TSO, and the Harris Hawk Optimization (HHO) algorithms were implemented in MATLAB software and tested on standard benchmark test functions and the pressure vessel design optimization problem. Through extensive simulations, the ITSO algorithm exhibits exceptional performance, outperforming the TSO and HHO algorithms across a variety of test functions. When restricting the three algorithms to a maximum of 100 iterations, the ITSO algorithm achieves considerably faster convergence on approximately 84.6% of the thirteen (13) test functions. Furthermore, an engineering design problem (the pressure vessel design problem) demonstrates the superior performance of the ITSO algorithm, yielding the best cost value of 5880.9471 as compared to 5885.3327 for the original TSO and 6393.0927 for the HHO. Given the ITSO algorithm's remarkable performance relative to the TSO and the HHO algorithms, the proposed ITSO is validated as an enhanced variant of the TSO. The ITSO can be applied to optimization problems in the electrical engineering field such as renewable energy integration and parameter tuning of control systems.*

Keywords: *Algorithm, Adaptive weight, Global Optimization, high-dimensional problem, Tuna Optimization Algorithm*

Article History
Received 30 July 2024
Received in revised form 19 August 2024
Accepted 22 August 2024

I. Introduction

In the domain of nature-inspired optimization algorithms, enhancing the ability to handle high-dimensional problems by improving convergence speed with accurate techniques is a pursuit that drives innovation and global performance improvements [1]. The pursuit to find more efficient algorithms has led to the development of numerous metaheuristic techniques which sometimes have some gaps, each aiming to address specific optimization challenges [2]-[4]. For example, the foraging behavior of pelicans has inspired the development of the pelican optimization algorithm (POA) for solving engineering design problems [5], the artificial gorilla troops optimizer inspired by gorillas to solve high-dimensional optimization problems [6], and so on.

Among these, Tuna Swarm Optimization (TSO) has proven to be a robust and effective method for solving complex optimization problems [7]. However, like many optimization algorithms, TSO has an inherent limitation in terms of handling high-dimensional problems. It converges slowly on such optimization problems and this can lead to the production of suboptimal solutions [8].

This paper introduces an improvement modification to the Tuna Swarm Optimization algorithm, which is term as the Improved Tuna Swarm Optimization (ITSO) algorithm, which focuses on balancing its exploration and exploitation mechanisms for efficient convergence from local optimal to global optimal solutions.

About the conventional optimization approaches, the ITSO algorithm introduces an adaptive fitness weight mechanism that dynamically adjusts the great impact of fitness values during the optimization process to improve the convergence speed. This adaptive fitness weight concept is aimed to facilitate faster convergence to optimal solutions, making it a promising avenue for addressing the TSO algorithm's slow convergence limitations [9].

To assess the worth of the effectiveness of the proposed ITSO algorithm, this study conducts a comprehensive assessment that includes both simulation-based comparisons and convergence analysis [10]-[11]. The proposed ITSO algorithm is implemented in MATLAB software, adhering to the same parameter settings as the original TSO algorithm [7]. Subsequently, performance results are compared with those of TSO and the Harris Hawks Optimization (HHO) algorithm, two well-established optimization techniques in the literature [7]-[11].

The experiments are designed with an interest in the algorithm's ability to handle high-dimensional problems with fast convergence to better solutions which will lead to efficient transition from local optimal to global optimal. Hence, attention is given to the problem dimensions and maximum number of iterations on each test function. Benchmark functions are employed to assess the performance of the ITSO algorithm in comparison to the original TSO and the HHO [7]-[12].

In addition to presenting comparative results, this paper provides detailed convergence curves, offering a visual representation of the convergence characteristics of the ITSO algorithm. These curves vividly illustrate the efficient convergence capabilities of the ITSO, emphasizing its potential in the field of optimization by addressing the need for efficient convergence to better solutions.

The impact of this research is not only to the advancement of optimization algorithms but also to emphasize the great impact of adaptive fitness weights in improving metaheuristic algorithms. Weight-based modifications make algorithms have a strong positive impact on obtaining better solutions in applications where faster convergence is a crucial requirement, including real-world problems in engineering [13], economics, and various other domains.

This work serves as a testament to the in-progress attempt to speed the boundaries of optimization, demonstrating that small yet innovative modifications can yield substantial improvements in algorithms' performance.

II. Methodology

This section presents the conventional tuna swarm optimization (TSO) algorithm, and a proposed modification to create an effective variant from the tuna swarm optimization algorithm to nullify premature convergence and enhance global performance for best solutions.

A. The Original Tuna Swarm Optimization (TSO) Algorithm

The Tuna Swarm Optimization (TSO) algorithm is a nature-inspired metaheuristic algorithm that derived its inspiration from the foraging behavior of tuna fish [7]. The foraging technique comprised of two stages detailed as follows.

The initial approach is spiral foraging, where tuna use a spiral formation during hunting. This method allows them to herd their prey into shallower waters, making it easier to capture. By adopting this spiral tactic, tuna effectively corral their prey and enhance their chances of a successful hunt.

The second method, known as parabolic foraging, involves each tuna following the one ahead, forming a parabolic pattern to effectively encircle its prey. The Tuna Swarm Optimization (TSO) algorithm is inspired by these natural foraging behaviors of tuna, particularly the spiral and parabolic strategies. By mimicking these tactics, the TSO algorithm enhances its optimization processes. The mathematical modeling of these behaviors is presented below:

Initialization:

Like many other nature-inspired metaheuristic algorithms, TSO commences the optimization process by randomly generating initial populations distributed uniformly across the search space using (1).

$$X_i^{int} = rand.(ub - lb) + lb, \quad i = 1, 2, \dots NP \quad (1)$$

where; x_i^{int} is the i^{th} individual, lb and ub are the lower and upper bounds of the search space, *NP* represents the number of tuna populations, and the rand is a uniformly distributed random vector with values ranging from 0 to 1.

Spiral foraging:

When faced with predators, small schooling fish such as sardines and herring adopt a dynamic formation, constantly changing their swimming direction to thwart predators' targeting efforts [14]-[15]. In contrast, tuna groups employ a tightly coiled spiral formation to pursue their prey. While most fish in the school may lack a strong sense of direction, they adjust their course to align with a small group of purposeful swimmers, ultimately forming a united hunting force [16]. Additionally, tuna schools engage in information exchange, with each tuna following the fish ahead, facilitating the sharing of information among neighboring tuna.

$$X_i^{t+1} = \begin{cases} \alpha_1.(X_{best}^t + \beta.|X_{best}^t - X_i^t|) + \alpha_2.X_i^t, i = 1 \\ \alpha_1.(X_{best}^t + \beta.|X_{best}^t - X_i^t|) + \alpha_2.X_{i-1}^t, i = 2,3,..NP \end{cases} \tag{2}$$

Also, when the most suitable member of the group cannot locate food, mindlessly trailing this individual during foraging is counterproductive. Thus, the idea of creating a random point within the search area as a reference for the spiral search was developed. This approach enables each group member to explore a broader area and enhances the group's ability to engage in global exploration. The precise mathematical model is outlined as follows:

$$X_i^{t+1} = \begin{cases} \alpha_1.(X_{best}^t + \beta.|X_{rand}^t - X_i^t|) + \alpha_2.X_i^t, i = 1 \\ \alpha_1.(X_{best}^t + \beta.|X_{rand}^t - X_i^t|) + \alpha_2.X_{i-1}^t, i = 2,3,..NP \end{cases} \tag{3}$$

$$\alpha_1 = a + (1-a).\frac{t}{t_{max}} \tag{4}$$

$$\alpha_2 = (1-a) - (1-a).\frac{t}{t_{max}} \tag{5}$$

$$\beta = e^{bl}.cos(2\pi b) \tag{6}$$

$$l = e^{3cos\left(\left(\left(t_{max}+\frac{1}{t}\right)-1\right)\pi\right)} \tag{7}$$

where; X_i^{t+1} represents the individual in the next iteration, X_{best}^t denotes the current best individual or optimal choice. α_1 and α_2 are coefficients that influence how individuals are inclined to move toward the best choice and their previous selection. a is a constant that determines the extent to which the individuals follow the best choice and the preceding selection in the initial phase. t represents the current iteration number, t_{max} is the maximum number of iterations allowed, and b is a randomly generated number uniformly distributed between 0 and 1.

Parabolic Foraging:

Apart from the spiral feeding pattern, tunas engage in cooperative feeding by adopting a parabolic formation. In this formation, they use a reference point, which is typically the location of their food. Furthermore, tunas actively search for food in their immediate surroundings. These dual feeding methods are executed together, each with an equal assumed likelihood of 50%. The mathematical model that describes this phenomenon can be outlined as follows:

$$X_i^{t+1} = \begin{cases} X_{best}^t + rand.(X_{best}^t - X_i^t) + TF.p^2.(X_{best}^t - X_i^t), & \text{if } r < 0.5 \\ TF.p^2 X_i^t, & \text{if } r \geq 0.5 \end{cases} \tag{8}$$

$$p = \left(1 - \frac{t}{t_{max}}\right)^{\frac{t}{t_{max}}} \tag{9}$$

where *TP* is a random number with a value of 1 or -1.

The above detail elaborated mathematical modeling of the foraging behavior of the tuna fish swarm in the aquatic environment is the major instrument behind the TSO algorithm. Just like other metaheuristic algorithms, the TSO algorithm follows systematic steps for effective implementation. The implementation procedure is presented in Fig. 1.

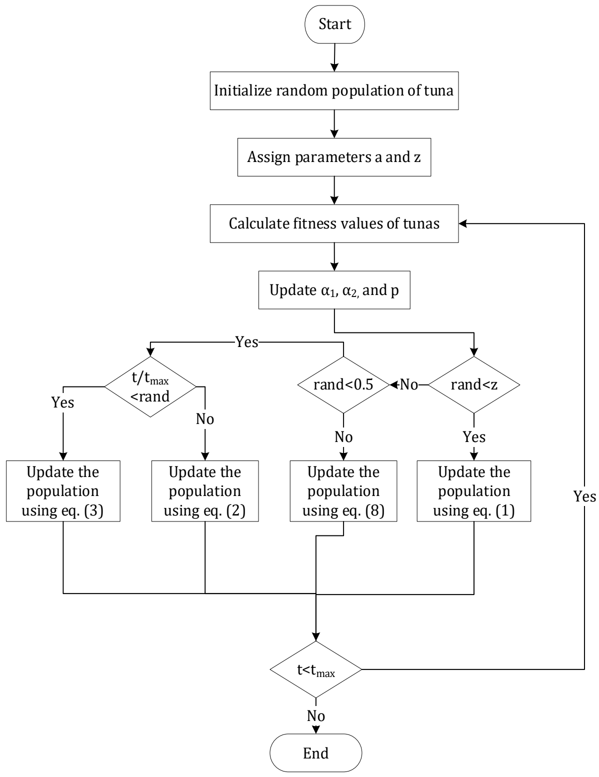

Fig. 1. Implementation Flowchart of TSO

B. Proposed Modification

The modification proposed in this work is to improve the performance of the TSO algorithm in terms of obtaining better solutions with faster convergence. This mainly involves the introduction of alternative weight calculation in the parabolic foraging phase. In this phase, the weight *p* is calculated using Eqn. (9). This approach of determining the value of *p* is mainly dependent on the maximum number of iterations and the iteration counter which gives a decreasing weight value. The dependence on the maximum number of iterations is the key cause of the slow convergence of the TSO in some cases (especially in solving high-dimensional problems).

To remedy this drawback, an adaptive fitness weight approach is proposed to replace the traditional one given in Eqn. (9) to enhance the algorithm's convergence speed to better solutions. Adaptive fitness weighting accelerates convergence by dynamically adjusting the importance of solutions based on their fitness, allowing the TSO to efficiently focus on promising regions, prevent premature convergence, and adapt to changing fitness landscapes, thus enhancing exploration, exploitation, and resource allocation that would ultimately improve the algorithm's convergence speed in various optimization problems to produce better global solutions [17]-[18]. The proposed calculation of weight, p, is presented in (10) below [19].

$$p = \frac{fit(X_i^t)}{\sum_{i=1}^{t} fit(X_i^t)} \tag{10}$$

where $fit(X_i^t)$ is the fitness of the individual tuna in iteration *t*. This adaptive weight is distinct from other techniques as it primarily depends on the individual fitness of the population to guide convergence. It is important to note that the weight is calculated at every iteration using Eqn. (10). The proposed ITSO algorithm implementation procedure is presented as follows:

1. *Generate the initial tuna population randomly*
2. *Assign the values of parameters a and z*
3. *Calculate the fitness of the tuna population.*
4. *Update the values of α_1, α_2, and p using (4), (5), and (10) respectively.*
5. *If rand<z, update the population using (1).*
6. *If rand<0.5, update the population using (8).*
7. *If t/t_{max}<rand, update the population using (3).*
8. *If t/t_{max}>rand, update the population using (2).*
9. *Repeat steps 3 to 8 if ($t<t_{max}$).*
10. *End and present the best fitness value as the optimal solution if ($t \geq t_{max}$).*

C. Experimental Test Setup

Test on Standard Benchmark Functions:

To evaluate the performance of the proposed ITSO algorithm, the same set of widely recognized benchmark functions utilized in [7] is employed for testing purposes. This collection includes 7 unimodal functions, 6 multimodal functions, and 10 complex multimodal functions. The unimodal functions, designated as F1–F7, have a single global optimal solution and are often used to assess an algorithm's ability to effectively exploit the local search space.

On the other hand, the multimodal functions, identified as F8–F13, feature multiple local optimal solutions, providing a challenge for the algorithm in terms of global exploration and the ability to avoid local optima. Detailed characteristics of these benchmark functions are available in Table I.

By testing the ITSO algorithm against these benchmark functions, its efficiency and robustness in both local and global optimization scenarios can be comprehensively assessed. This rigorous testing ensures that the algorithm's strengths and potential weaknesses are thoroughly examined, providing valuable insights into its overall performance.

TABLE I
CHARACTERISTICS OF BENCHMARK FUNCTIONS

Function	Name	Dim	Range	Global Solution
F1	Sphere	30	[-100, 100]	0
F2	Schwefel 2.22	30	[-10, 10]	0
F3	Schwefel 1.2	30	[-100, 100]	0
F4	Schwefel 2.21	30	[-100, 100]	0
F5	Rosenbrock	30	[-30, 30]	0
F6	Step	30	[-100, 100]	0
F7	Quartic	30	[-1.28, 1.28]	0
F8	Schwefel 2.26	30	[-500, 500]	-418.9829*D
F9	Rastrigin	30	[-5.12, 5.12]	0
F10	Ackley	30	[-32, 32]	8.881E-16
F11	Griewank	30	[-600, 600]	0
F12	Penalized	30	[-50, 50]	0
F13	Penalized 2	30	[-50, 50]	0

The identical simulation parameter configurations employed in the TSO algorithm [7] have been applied in this test for conducting a fare comparison. The parameter settings are presented in Table II.

TABLE II
PARAMETER SETTINGS

Algorithm	Parameters
HHO	~
TSO	a=0.7, z=0.05
ITSO	a=0.7, z=0.05

The simulation tests were run in MATLAB R2019a software using an HP Pavilion laptop computer with the following specifications: 64-bit Windows operating system, AMD A8-6410 APU processor with a clock speed of 2.00GHz, and installed RAM of 4GB (3.43 GB usable).

Test on Engineering Design Problem (Pressure Vessel Design Problem):

The issue of pressure vessel design presented in Fig. 2 is a widely recognized test case that aims to minimize the overall expenses, encompassing expenses related to shaping, materials, and welding. This problem involves four distinct factors: the thickness of the vessel (Ts, represented as x_1), the thickness of the head (x_2), the inner diameter (R, represented as x_3), and the length of the cylindrical cross-section of the vessel (L, represented as x_4). The problem is outlined as follows:

$$\begin{aligned} \min. f(x_1, x_2, x_3, x_4) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ +3.1661x_1^2x_4 + 19.84x_1^2x_3 \end{aligned} \tag{11}$$

Subject to:

$$\begin{cases} g_1(x) = -x_1 + 0.0193x_3 \leq 0, \\ g_2(x) = -x_2 + 0.00954x_3 \leq 0, \\ g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^2 + 1{,}296{,}000 \leq 0, \\ g_4(x) = x_4 - 240 \leq 0, \end{cases} \tag{12}$$

Variable ranges: $1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$, $10 \leq x_3, x_4 \leq 200$

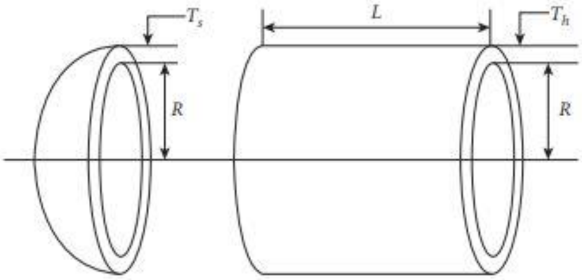

Fig. 2. Schematic of the pressure vessel design problem

III. Results and Discussion

This section presents the simulation results of the ITSO algorithm, comparing its performance to the original TSO algorithm and the HHO algorithm. The evaluation is conducted using benchmark functions and the pressure vessel design problem [20]-[21].

A. Results on standard benchmark functions

To effectively assess the ITSO algorithm, its performance is compared across different problem dimensions: 30, 100, and 500. These comparisons are detailed in Table III, Table IV, and Table V, respectively, to evaluate the algorithm's capability in handling high-dimensional problems. The analysis of the results is presented following each table, providing insights into the ITSO algorithm's effectiveness and robustness across varying problem complexities. This comprehensive evaluation aims to highlight the algorithm's strengths and potential areas for improvement when dealing with different scales of problem dimensions.

Table III presents a comparative analysis of the results obtained from the three algorithms, evaluated using a problem dimension of 30 across thirteen benchmark functions, labeled F1 to F13. The findings reveal that the Improved Tuna Swarm Optimization (ITSO) algorithm significantly outperforms both the original Tuna Swarm Optimization (TSO) and the Harris Hawks Optimization (HHO) algorithms in terms of average performance and standard deviation. This exceptional performance, however, is not consistent across all benchmark functions, as it falls short in functions F1, F3, F9, F10, and F11.

In the cases of F1 and F3, both the ITSO and TSO algorithms were able to achieve the global optimum solutions, demonstrating comparable effectiveness in these specific instances. Furthermore, all three algorithms, including ITSO, TSO, and HHO, succeeded in attaining global average and standard deviation solutions for functions F9, F10, and F11. Despite these exceptions, the overall performance highlighted in Table III suggests that the ITSO algorithm is generally superior in tackling optimization problems across various domains when compared to its counterparts, TSO and HHO.

TABLE III
COMPARISON OF RESULTS ON F1-F13 FOR 1000 ITERATIONS WITH 30D

Function		ITSO	TSO	HHO
F1	Av	**0.00E+00**	**0.00E+00**	3.92E−193
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	Av	**0.00E+00**	1.47E−235	3.54E−102
	Std	**0.00E+00**	**0.00 E+00**	1.17E−101
F3	Av	**0.00E+00**	**0.00E+00**	1.98E−155
	Std	**0.00E+00**	**0.00E+00**	1.09E−154
F4	Av	**0.00E+00**	2.39E−236	3.11E−98
	Std	**0.00E+00**	**0.00E+00**	1.14E−97
F5	Av	**5.35E-05**	1.22E−04	9.96E−04
	Std	**5.33E-05**	3.16E−04	1.09E−03
F6	Av	**2.59E-08**	1.77E−08	9.32E−06
	Std	**7.26E-08**	9.08E−08	1.44E−05
F7	Av	**3.66E-05**	1.15E−04	3.67E− 05
	Std	**3.19E-05**	7.56E−05	3.20E− 05
F8	Av	**-1.25E+04**	-1.26E+04	−1.26
	Std	**1.99E-06**	1.64E−06	E+04
				8.89E−02
F9	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	Av	**8.88E-16**	**8.88E−16**	**8.88E−16**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	Av	**6.26E-11**	3.16E−10	8.06E−07
	Std	**6.96E-11**	8.13E−10	1.06E−06
F13	Av	**3.23E-10**	1.93E−09	5.48E−06
	Std	**5.25E-10**	4.41E−09	5.87E−06

This comparative performance evaluation emphasizes the potential of the ITSO algorithm as a robust solution for optimization challenges. The ability to outperform both the original TSO and HHO in most scenarios showcases its effectiveness and adaptability in diverse applications.

To evaluate the performance of the ITSO algorithm with an increased problem dimension of 100, a comparative analysis was conducted against the original TSO and the HHO algorithms, as illustrated in Table IV. The results indicated that the ITSO algorithm generally outperformed both the TSO and HHO algorithms across most benchmark functions. The only exception was in function F6, where the HHO algorithm achieved the best performance. Nonetheless, the ITSO showed improvement in F6 compared to the original TSO.

For functions F1 and F3, both the ITSO and TSO algorithms were able to reach the global optimum solutions. Additionally, all three algorithms - ITSO, TSO, and HHO - successfully identified the global solutions for functions F9, F10, and F11. This assessment highlights the effectiveness of the ITSO algorithm in adapting to increased problem dimensions and demonstrates its competitive edge in optimization tasks, further solidifying its role as a strong candidate among the algorithms evaluated.

TABLE IV
COMPARISON OF RESULTS ON F1-F13 FOR 1000 ITERATIONS WITH 100D

Function		ITSO	TSO	HHO
F1	Av	**0.00E+00**	**0.00E+00**	5.76E−190
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	Av	**0.00E+00**	1.96E−231	3.05E−100
	Std	**0.00E+00**	**0.00 E+00**	1.30E−99
F3	Av	**0.00E+00**	**0.00E+00**	2.81E−145
	Std	**0.00E+00**	**0.00E+00**	1.54E−144
F4	Av	**0.00E+00**	1.49E−229	1.30E−97
	Std	**0.00E+00**	**0.00E+00**	4.59E− 97
F5	Av	**2.53E-04**	1.15E−01	3.74E−03
	Std	**3.55E-04**	4.32E−01	5.74E−03
F6	Av	1.02E-03	1.08E−03	**3.41E−05**
	Std	7.30E-04	2.11E−03	**5.76E−05**
F7	Av	**3.54E-05**	1.17E−04	4.11E−05
	Std	**3.92E-05**	1.62E−04	6.24E−05
F8	Av	**-4.19E+04**	−4.19E+04	−4.19 E+04
	Std	**5.13E-02**	8.33E−02	3.21E−01
F9	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	Av	**8.88E−16**	**8.88E−16**	**8.88E−16**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	Av	**1.69E-07**	3.03E−06	3.47E−07
	Std	**1.18E-07**	8.46E−06	5.00E−07
F13	Av	**1.16E-05**	1.44E−04	1.19E−05
	Std	**1.67E-05**	2.32E−04	1.74E−05

To get a more convincing performance of the ITSO algorithm, the problem dimension was further increased to 500 and the result is presented in Table V.

TABLE V
COMPARISON OF RESULTS ON F1-F13 FOR 1000 ITERATIONS WITH 500D

Function		ITSO	TSO	HHO
F1	Av	**0.00E+00**	**0.00E+00**	6.83E−192
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	Av	**0.00E+00**	1.24E−230	4.93E−96
	Std	**0.00E+00**	**0.00 E+00**	2.70E−95
F3	Av	**0.00E+00**	**0.00E+00**	1.08E−87
	Std	**0.00E+00**	**0.00E+00**	5.91E−87
F4	Av	**3.15E-238**	2.22E−228	7.26E−94
	Std	**0.00E+00**	**0.00E+00**	3.97E−93
F5	Av	**2.38E-03**	9.10E−01	1.53E−02
	Std	**9.78E-03**	1.41 E+00	1.80E−02
F6	Av	1.67E-01	1.68E−01	**1.01E−04**
	Std	2.83E-01	2.27E−01	**1.27E−04**
F7	Av	**3.33E-05**	1.20E−04	3.74E−05
	Std	**3.02E-05**	1.13E−04	3.03E−05
F8	Av	**-2.09E-05**	−2.09E+05	−2.09 E+5
	Std	**2.05E-01**	7.35E−01	1.36 E+00
F9	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	Av	**8.88E−6**	**8.88E−06**	**8.88E−06**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	Av	9.67E-06	2.02E−05	**2.78E−07**
	Std	8.44E-06	5.06E−05	**3.95E−07**
F13	Av	7.16E-04	4.21E−03	**4.36E−05**
	Std	1.33E-03	1.21E−02	**7.01E−05**

Table V displays a performance pattern similar to that observed in Table IV, with notable exceptions for functions F12 and F13. When the problem dimension was significantly increased to 500, the performance of the ITSO algorithm in F12 and F13 was adversely affected, leading to its underperformance compared to the HHO algorithm. Despite this, the ITSO algorithm still demonstrated a significant advantage over the original TSO algorithm, achieving competitive results against the HHO algorithm for the test functions F12 and F13. Throughout all three tables, the ITSO consistently exhibited superior performance relative to the other algorithms, as indicated by its average (Av) and standard deviation (Std) values.

In addition to enhancing overall performance, a key objective of the proposed ITSO algorithm is to improve convergence speed. To evaluate this aspect, simulations of the three algorithms were conducted using the default problem dimension of 30, but with a reduced number of iterations set to 100. The results of this simulation are summarized in Table VI, which further illustrates the convergence characteristics of each algorithm. This focus on convergence speed, alongside performance improvements, highlights the effectiveness of the ITSO algorithm in optimization tasks.

TABLE VI
COMPARISON OF RESULTS ON F1-F13 FOR 100 ITERATIONS WITH 30D

Function		ITSO	TSO	HHO
F1	Av	**0.00E+00**	5.43E-47	3.70E-23
	Std	**0.00E+00**	2.97E-46	1.38E-22
F2	Av	**1.87E-134**	1.71E-25	5.62E-13
	Std	**1.02E-133**	7.21E-25	1.33E-12
F3	Av	**2.81E-111**	2.92E-45	2.50E-14
	Std	**1.54E-110**	1.25E-44	8.82E-14
F4	Av	**1.92E-81**	1.21E-24	4.58E-12
	Std	**1.05E-80**	4.24E-24	1.50E-11
F5	Av	**3.23E+00**	8.47E+00	5.87E+00
	Std	**7.07E+00**	1.10E+01	3.34E+01
F6	Av	**3.97E-02**	2.65E-01	3.58E-01
	Std	**4.30E-02**	2.94E-01	4.72E-01
F7	Av	**3.36E-04**	1.19E-03	8.18E-04
	Std	**3.42E-04**	9.26E-04	8.35E-04
F8	Av	**-1.256E+04**	-12561E+04	-1.2354E+04
	Std	**3.9769E+00**	1.3465E+01	768.356E+00
F9	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	Av	**8.88E-16**	**8.88E-16**	1.636E-13
	Std	**0.00E+00**	**0.00E+00**	3.2832E-13
F11	Av	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	Av	**3.540E-04**	4.224E-03	5.4757E-04
	Std	**4.236E-04**	8.701E-03	5.6237E-04
F13	Av	**1.210E-02**	1.664E-02	3.2811E-02
	Std	**1.804E-02**	2.5247E-02	3.7605E-02

The results presented in Table VI indicate that the ITSO algorithm outperforms both the original TSO and the HHO algorithms across functions F1, F2, F3, F4, F5, F6, F7, F8, F12, and F13. Additionally, for functions F9 and F11, all three algorithms successfully achieved global optimal solutions. In function F10, both the ITSO and original TSO algorithms also attained the global solution. Overall, the ITSO algorithm demonstrates superior performance compared to the other algorithms, particularly regarding fast convergence to improved solutions.

The convergence characteristics of the algorithms are further illustrated in the subsequent curves, which provide a more detailed comparison of their performance over the iterations. This analysis reinforces the effectiveness of the ITSO algorithm in achieving quicker convergence while maintaining a competitive edge in optimization tasks. By consistently delivering better results across multiple functions, the ITSO algorithm stands out as a promising approach for optimization challenges.

B. Convergence Characteristics

The curves illustrated below represent the convergence characteristics of the three algorithms (ITO, TSO, and HHO) from simulations with a maximum iteration of 100. These compare the algorithms' convergence properties concerning fast convergence to better solutions. Figure 3-15 shows the thirteen (13) standard benchmark functions and hence presents the convergence characteristics of the three (3) algorithms on all the test functions. The curves are extensively elaborated for clear interpretations.

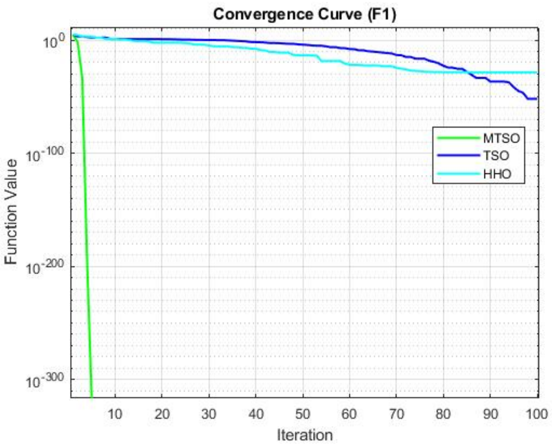

Fig. 3. Convergence on Sphere

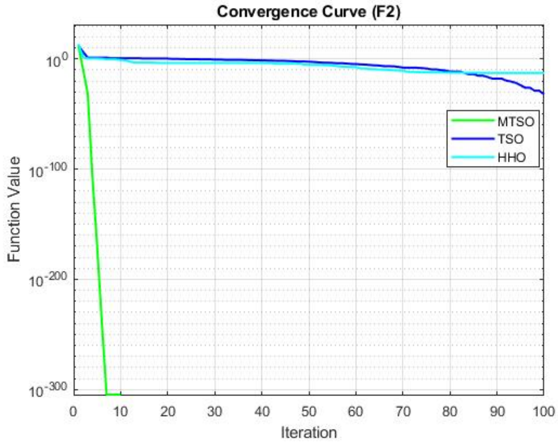

Fig. 4. Convergence on Schwefel 2.22

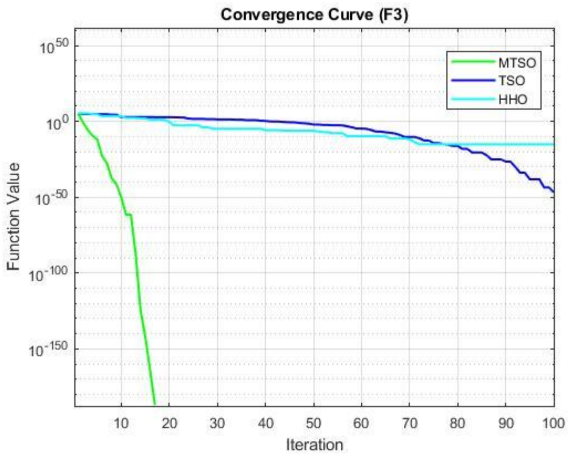

Fig. 5. Convergence on Schwefel 1.2

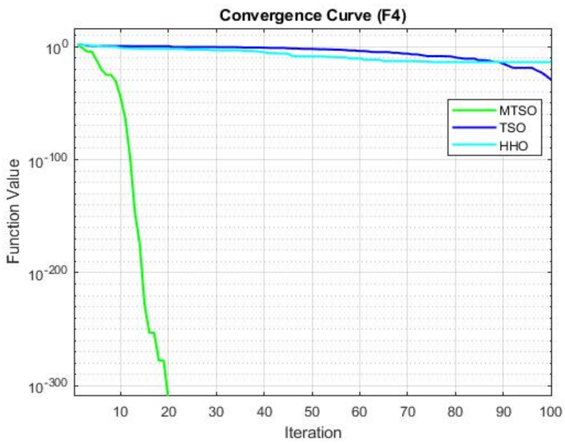

Fig. 6. Convergence on Schwefel 2.21

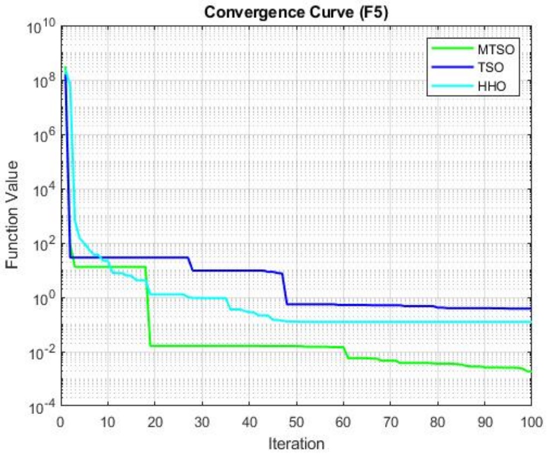

Fig. 7. Convergence on Rosenbrock

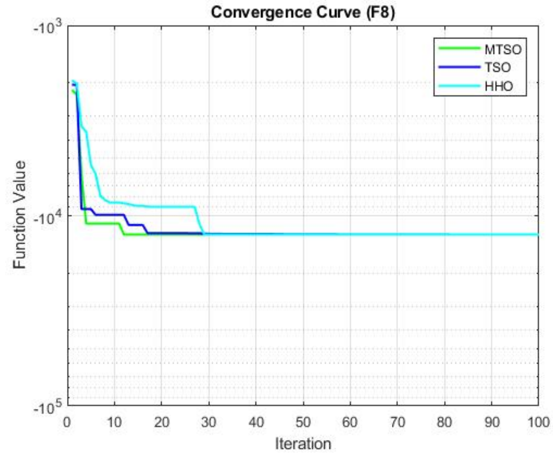

Figure 10: Convergence on Schwefel 2.26

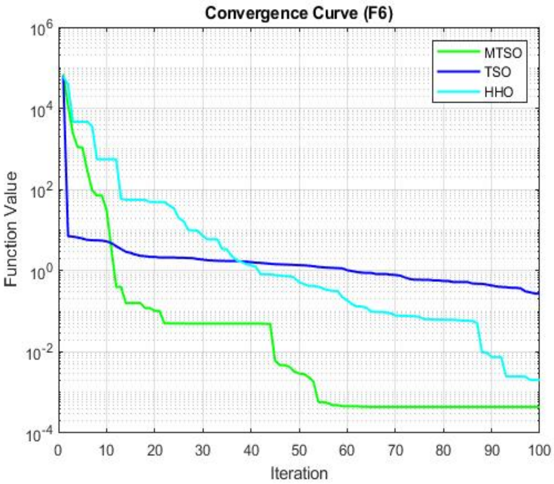

Fig. 8. Convergence on Step

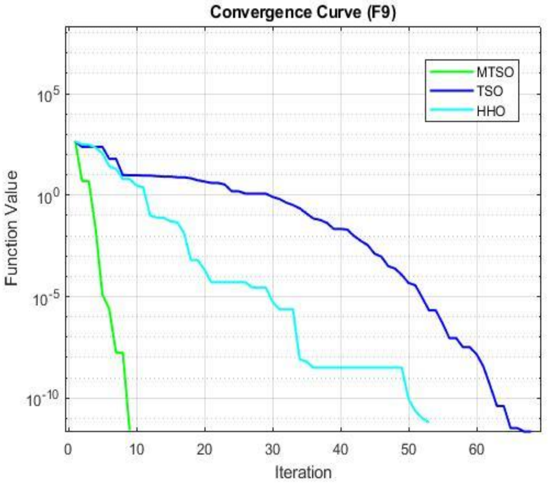

Fig. 11. Convergence on Rastrigin

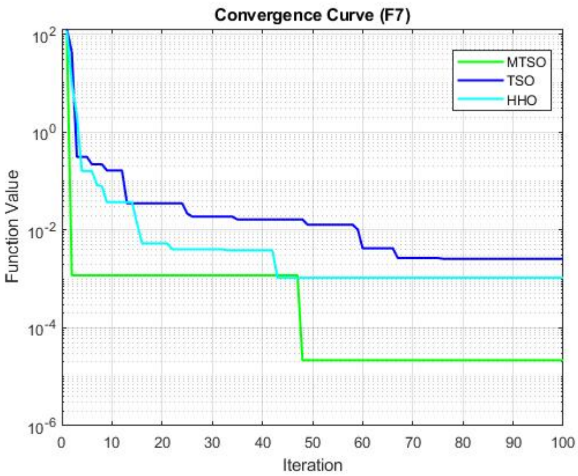

Fig. 9. Convergence on Quartic

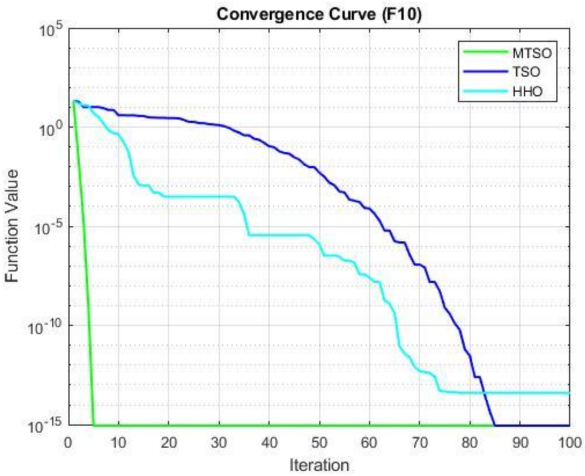

Fig. 12. Convergence on Ackley

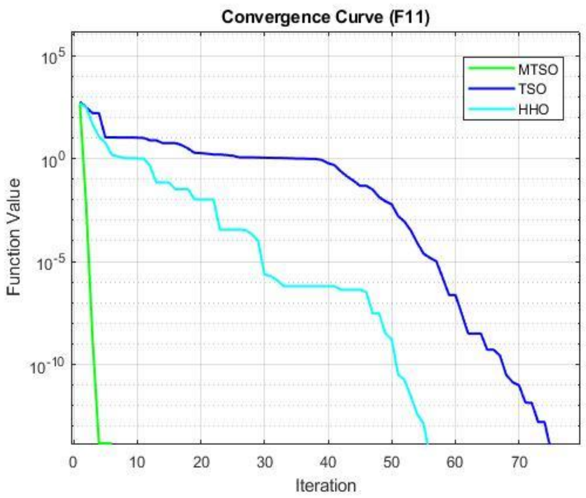

Fig. 13. Convergence on Griewank

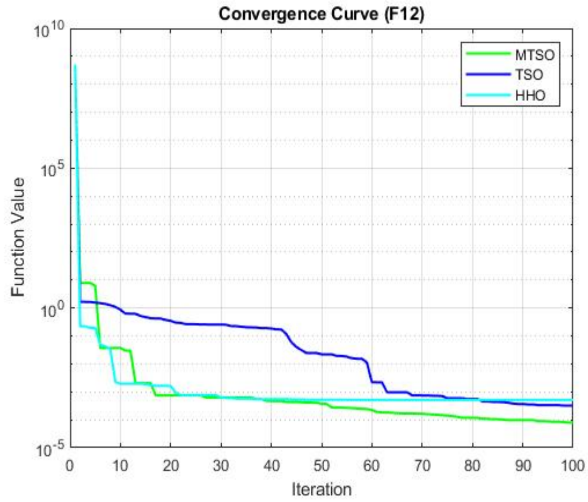

Fig. 14. Convergence on Penalized

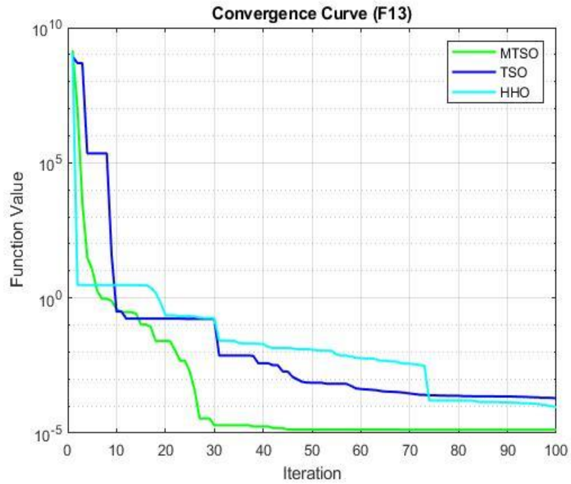

Fig. 15. Convergence on Penalized 2

From the convergence curves, the presumed ITSO algorithm, represented by the green curves, showed exceptionally better convergence characteristics in F1, F2, F3, F4, F5, F6, F7, F9, F10, F11, and F13. These represent about 84.6% of the 13 functions tested. In the case of functions F8 and F12, the three algorithms produced very close convergence characteristics with the ITSO algorithm slightly better at the final solutions. Generally, there has been great improvement in the convergence speed of the modification without deteriorating the effective exploration to better optimal solutions.

C. Performance on Engineering Design Problem (Pressure Vessel Design Problem)

The performance of a typical engineering design problem, the pressure vessel design problem, is assessed in Table VII.

TABLE VII
RESULTS COMPARISON ON PRESSURE VESSEL DESIGN PROBLEM

Algorithm	HHO	TSO	ITSO
X_1	0.9459	0.7782	0.7782065
X_2	0.4471	0.3846	0.3830365
X_3	46.8513	40.3196	40.31963
X_4	125.468	199.999	200
Optimal Cost	6393.0927	5885.3327	5880.9471

The ITSO algorithm simulation result on the design problem is compared to those of the original TSO and the HHO algorithms reported in the literature. This test is to assess the potential of the ITSO algorithm on engineering optimization problems, and it is obvious the ITSO algorithm outperformed the other algorithms with the lowest cost value of 5880.9471 against that of 5885.3327 and 6393.0927 for the TSO and HHO algorithms respectively.

The findings indicate that utilizing the ITSO algorithm could lead to more efficient and reliable solutions in optimization tasks, reinforcing its position as a preferred choice among the evaluated algorithms. Overall, this analysis underscores the advancements made by the proposed ITSO algorithm in the field of optimization.

IV. Conclusion and Recommendation

A modified version of the TSO algorithm called the Improved Tuna Swarm Optimization (ITSO) algorithm is proposed. An adaptive fitness weight strategy is used to represent the formal strategy for calculating the weight value to arrive at quick optimal convergence to better the solutions of the conventional approach which is the Tuna Swarm Optimization (TSO) algorithm. Simulation tests using the same parameter settings in the literature of the original TSO algorithm showed exceptional performance of the ITSO over the TSO and the HHO algorithms. A test of the three algorithms on the same 13 test functions showed that the ITSO algorithm has a faster convergence speed and better performance as compared to the TSO and the HHO algorithms on about 84.6% of the 13 benchmark functions. A final test on a typical engineering design problem, that is the pressure vessel design problem showed a better performance in the benefit of the ITSO algorithm with the best cost value of 5880.9471 compared to those of the TSO and the HHO that had 5885.3327 and 6393.0927 respectively. With the good performance exhibited by the ITSO algorithm in terms of fast convergence and better solutions, it is recommended that researchers explore the potential of the proposed ITSO algorithm in solving optimization problems that require fewer iterations and fast convergence. Additional works focus on looking into the application of the ITSO algorithm in solving engineering optimization problems.

Author Contributions

Author 1: Research conceptualization, Supervision, draft review, and editing; Author 2: Conducts simulation test, analysis, and interpretation of results; Author 3: Writing – original draft preparation.

Conflict of Interest

The authors declare no conflict of interest in the publication process of the research article.

References

[1] A.-F. Seini Yussif, E. Twumasi, and E. A. Frimpong, "Performance Enhancement of Elephant Herding Optimization Algorithm Using Modified Update Operators," *J. Nas. Tek. Elektro*, vol. 2, no. 30, 2023, doi: 10.25077/jnte.v12n2.1124.2023.

[2] A. Ramadan, S. Kamel, M. M. Hussein, and M. H. Hassan, "A new application of chaos game optimization algorithm for parameters extraction of three diode photovoltaic model," *IEEE Access*, vol. 9, pp. 51582–51594, 2021, doi: 10.1109/ACCESS.2021.3069939.

[3] O. S. Elazab, H. M. Hasanien, I. Alsaidan, A. Y. Abdelaziz, and S. M. Muyeen, "Parameter estimation of three diode photovoltaic model using grasshopper optimization algorithm," *Energies*, vol. 13, no. 2, 2020, doi: 10.3390/en13020497.

[4] T. Yuvaraj *et al.*, "Optimal integration of capacitor and distributed generation in distribution system considering load variation using bat optimization algorithm," *Energies*, vol. 14, no. 12, 2021, doi: 10.3390/en14123548.

[5] P. Trojovský and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications," *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22030855.

[6] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Int. J. Intell. Syst.*, vol. 36, no. 10, pp. 5887–5958, 2021, doi: 10.1002/int.22535.

[7] L. Xie, T. Han, H. Zhou, Z. R. Zhang, B. Han, and A. Tang, "Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization," *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/9210050.

[8] A. Awad, S. Kamel, M. H. Hassan, and M. F. Elnaggar, "An Enhanced Tuna Swarm Algorithm for Optimizing FACTS and Wind Turbine Allocation in Power Systems," *Electr. Power Components Syst.*, vol. 0, no. 0, pp. 1–16, 2023, doi: 10.1080/15325008.2023.2237011.

[9] Z. Yan, J. Yan, Y. Wu, S. Cai, and H. Wang, "A novel reinforcement learning based tuna swarm optimization algorithm for autonomous underwater vehicle path planning," *Math. Comput. Simul.*, vol. 209, pp. 55–86, 2023, doi: 10.1016/j.matcom.2023.02.003.

[10] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Syst.*, vol. 191, p. 105190, 2020, doi: 10.1016/j.knosys.2019.105190.

[11] Z. Garip, "Parameters estimation of three-diode photovoltaic model using fractional-order Harris Hawks optimization algorithm," *Optik (Stuttg).*, vol. 272, no. December 2022, p. 170391, 2023, doi: 10.1016/j.ijleo.2022.170391.

[12] M. A. Al-Betar *et al.*, "A hybrid Harris Hawks optimizer for economic load dispatch problems," *Alexandria Eng. J.*, vol. 64, pp. 365–389, 2023, doi: 10.1016/j.aej.2022.09.010.

[13] A.-F. Seini Yussif and T. Seini, "Improved F-parameter Mountain Gazelle Optimizer (IFMGO): A Comparative Analysis on Engineering Design Problems," *Int. Res. J. Eng. Technol.*, no. July, pp. 810–816, 2023, [Online]. Available: www.irjet.net.

[14] X. Wang *et al.*, "Modeling collective motion for fish schooling via multi-agent reinforcement learning," *Ecol. Modell.*, vol. 477, no. December 2022, p. 110259, 2023, doi: 10.1016/j.ecolmodel.2022.110259.

[15] D. T. de Kerckhove and B. J. Shuter, "Predation on schooling fish is shaped by encounters between prey during school formation using an Ideal Gas Model of animal movement," *Ecol. Modell.*, vol. 470, no. May, p. 110008, 2022, doi: 10.1016/j.ecolmodel.2022.110008.

[16] C. Kumar and D. Magdalin Mary, "A novel chaotic-driven Tuna Swarm Optimizer with Newton-Raphson method for parameter identification of three-diode equivalent circuit model of solar photovoltaic cells/modules," *Optik (Stuttg).*, vol. 264, no. May, p. 169379, 2022, doi: 10.1016/j.ijleo.2022.169379.

[17] V. Christelis, G. Kopsiaftis, R. G. Regis, and A. Mantoglou, "An adaptive multi-fidelity optimization framework based on co-Kriging surrogate models and stochastic sampling with application to coastal aquifer management," *Adv. Water Resour.*, vol. 180, no. January, p. 104537, 2023, doi: 10.1016/j.advwatres.2023.104537.

[18] Q. Wang, Q. Yang, S. He, Z. Shi, and J. Chen, "AsyncFedED: Asynchronous Federated Learning with Euclidean Distance based Adaptive Weight Aggregation," 2022, [Online]. Available: http://arxiv.org/abs/2205.13797.

[19] A. J. O. E. A. T. I.-L. ; Kwegyir, D. A. Frimpong, and E. A. Opoku, "Modified Local Leader Phase Spider Monkey Optimization Algorithm," ADRRI J. Eng. Technol., vol. 5, no. 2, pp. 1–18, 2021

[20] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Appl. Intell.*, vol. 51, no. 3, pp. 1531–1551, 2021, doi: 10.1007/s10489-020-01893-z.

[21] F. A. Hashim and A. G. Hussien, "Snake Optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Syst.*, vol. 242, p. 108320, 2022, doi: 10.1016/j.knosys.2022.108320.