International Journal of Innovative Information Systems & Technology Research 14(3):206-231, July-Sept, 2026

© SEAHI PUBLICATIONS, 2026 www.seahipublications.org ISSN: 2467-8562

DOI:10.5281/zenodo.22022147

Enhanced Marine Predator Optimization Algorithm: A Modified Version of Marine Predator Optimization Algorithm

Okwuokenye. Z. Chijioke[1], Isaac. I. Alabi[2], Abel E. Airoboman[3], Abba M. Adua[4] & Usman Isah Yusuf [5]

[1,2,3&4]Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna, Nigeria

[5]Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria

*** Corresponding author E-mail: engrusmanisah@gmail.com**

ABSTRACT

This study introduces an improved version of the Marine Predator Algorithm (eMPA). The original Marine Predator Algorithm (MPA) is based on the common foraging patterns of ocean predators, specifically utilizing Brownian motion, combined with an optimal encounter rate policy that models' biological interactions between predators and prey. MPA is designed to mimic the natural principles governing efficient foraging and prey encounters in marine environments. However, it struggles with maintaining a proper balance between exploration and exploitation, leaning more towards exploration, which increases the risk of getting stuck in local optima. To mitigate this issue, the enhanced version (eMPA) incorporates a Levy flight-inspired probability distribution to strengthen the exploration capabilities during the search process. The developed eMPA was also evaluated using 15 standard benchmark functions using best, mean, worst, standard deviation and rank as performance metrics. Simulation was performed using MATLAB R2022b and the results obtained was compared with other metaheuristic algorithms such as the original MPA, PSO and GWO. The choice of these algorithms was influenced by how best is their solution is near global optimal and their good convergence ability. Simulation result shows eMPA converges faster in 3 functions (F2, F3 and F4) out of 5 which is 60% of the functions over PSO and MPA and GWO. Also, in terms of moderate dimensional benchmark functions, it is observed that eMPA converges faster in 2 functions (F6 and F9) out of 5 which is 40% of the functions, MPA converges in function (F6) which is 20%, PSO converges faster in 3 functions (F6, F8 and F10) which is 60% and GWO converges in 2 functions (F1 and F7) which is 40% of the moderate-dimensional functions. The superior performance of PSO over eMPA and the other algorithms is attibuted to its high exploitation ability over exploration. These results demonstrate the importance of maintaining an appropriate exploration–exploitation balance: excessively rapid convergence may lead to premature convergence and loss of population diversity, whereas controlled convergence can provide better global-search capability. Finally, in terms of multimodal functions is also presented from Fig 14 to Fig 18. From these plots it is observed that eMPA converges faster in 3 functions (F11, F12 and F15) out of 5 which is 60% of the functions, MPA converges in 2 function (F11 and F12) which is 40%, PSO converges faster in 2 functions (F11 and F13) which is 20% of the functios and GWO converges in 1 functions (F11 and F12) which is 40% of the multimodal-dimensional functions. In F14 the high exploitation ability of PSO and GWO shows that both are in a straight horizontal line while eMPA and MPA takes longer number of iterations to converge due to high level of exploration over exploitation. These result also shows the superior performance of eMPA over the other algorithms.

Keywords: Marine Predator Algorithm, Metaheuristic, Stochastic Optimization, Global Optimization, Levy Flight, Swarm Intelligent

1. INTRODUCTION

In recent times, algorithms inspired by natural phenomena have garnered significant interest due to their success in addressing diverse complex optimization challenges and their capacity to deliver competitive outcomes [1]. Optimization can be understood as a structured approach to finding the most appropriate solution for a specific problem, ensuring that predefined constraints are met and resources are utilized effectively. Deterministic optimization algorithms generally produce the same solution when identical initial conditions and parameters are provided. In contrast, stochastic optimization techniques incorporate random processes into their search mechanisms, meaning that different runs may produce different solutions even when the initial conditions are unchanged [2]. Modern universal optimization techniques, computational intelligence, and soft-computing methods increasingly employ metaheuristic strategies derived from natural phenomena. Typically, optimization algorithms are classified into two main categories: deterministic methods and stochastic methods [3].

Earlier research has further divided stochastic optimization methods into heuristic and metaheuristic categories. Heuristic techniques tend to be tailored for specific types of optimization problems and are highly dependent on the problem context. Conversely, metaheuristic algorithms are generally problem-agnostic and can be applied across a wide variety of optimization challenges [4]. A key feature of metaheuristics is their capacity to maintain a balance between exploration (diversification) and exploitation (intensification). Exploration allows the algorithm to search different regions of the solution space to find promising candidates, while exploitation refines the search around high-quality solutions to enhance their fitness. Achieving an appropriate balance between these two aspects increases the likelihood of finding the global optimum and prevents early convergence to local optima [5]. Many metaheuristic strategies are inspired by natural, biological, physical, or social phenomena, and have been successfully used to address numerous complex optimization problems [6].

Particle Swarm Optimization (PSO) is a prominent example of a metaheuristic algorithm inspired by natural phenomena, modelled after the coordinated movement and social interactions seen in bird flocks and fish schools [7]. Although PSO is effective, it has some limitations, notably its relatively limited ability to perform local searches. This can lead the algorithm to become trapped in suboptimal solutions, diminishing the quality of convergence and introducing uncertainty into the results [8]. Several other algorithms inspired by biological behaviours have also been developed. For instance, the Firefly Algorithm (FA) mimics the flashing and attraction behaviours of fireflies [9], while Cat Swarm Optimization (CSO) is based on the hunting, tracking, and stalking actions of cats [10]. The Bat Algorithm (BA) models echolocation used by microbats for navigation and prey detection [11], and Ant Colony Optimization (ACO) draws inspiration from the foraging and route selection behaviours of ant colonies [12]. Similarly, the Dream Optimization Algorithm (DOA) is inspired by the human dreaming process, integrating features like memory retention, forgetting, and self-organization into the optimization process [13]. To enhance the balance between exploration and exploitation, DOA uses memory mechanisms, strategies for forgetting and replenishing information, and dream-sharing techniques. These features aim to improve the algorithm's ability to escape local optima while maintaining effective search capabilities.

Previous research has shown that the MPA exhibits competitive features when compared to several well-known metaheuristic algorithms such as Particle Swarm Optimization (PSO) [14], Grey Wolf Optimization (GWO) [15], Sparrow Search Algorithm (SSA) [16], Differentiated Creative Search (DCS) [17], Great Wall Construction Algorithm (GWCA) [18], and Whale Optimization Algorithm (WOA) [19]. The benefits reported include strong convergence abilities, enhanced search efficiency, stability, adaptability, robustness, and a lower sensitivity to the initial setting of control parameters. However, the Dream Optimization Algorithm (DOA) may still face challenges in maintaining a proper balance between exploration and exploitation. This issue can be caused by the ongoing influence of control parameters, the nature of the optimization search space, and limited information about the problem landscape, which may lead to premature convergence or inadequate exploration of promising areas.

To overcome these challenges, the research proposed an improved Marine Predator Algorithm (eMPA) by embedding a Lévy flight mechanism into the traditional MPA. This Lévy-flight-based approach aims to boost the global search ability of MPA and facilitate its escape from local optima. The effectiveness of the eMPA was tested on ten standard benchmark functions, and its performance was compared to that of the original MPA, White Shark Optimization (WSO), Seagull Optimization Algorithm (SOA), Sunflower Optimization (SFO), Golden Jackal Optimization (GJO), and African Vultures Optimization Algorithm (AVOA). The results of the comparisons indicated that the enhanced eMPA outperformed the other algorithms in optimization tasks. The main innovation of this study lies in the integration of Lévy flight into the traditional MPA to improve its search efficiency and overall optimization performance.

Metaheuristic algorithms can be grouped into four categories based on their underlying principles: evolution-inspired, physics-inspired, human-inspired, and swarm intelligence-based methods [20]. Figure 1 illustrates this classification [21].

Evolution-inspired algorithms draw upon natural evolutionary processes. They typically initiate with a randomly generated set of candidate solutions, and then improve these solutions through mechanisms such as mutation, crossover, and selection, using the best solutions to generate new candidates. Notable examples include Genetic Algorithms (GA), which are modelled after Darwinian evolution, as well as Evolution Strategies (ES), Genetic Programming (GP), Tabu Search, and Differential Evolution (DE) [20].

Physics-inspired algorithms are modelled on physical laws governing the universe. They also begin with a random initial population and refine solutions iteratively to find optimal results. Examples include Simulated Annealing (SA) and Harmony Search [22].

Nomenclature & Symbols			
MPA	Marine Predator Algorithm	eMPA	Enhance Marine Predator Algorithm
PSO	Particle Swarm Optimization	GA	Genetic Algorithm
ACO	Ant Colony Optimization	ES	Evolution Strategy
GP	Genetic Programming	DE	Differential Evolution
SA	Simulated Annealing	TLBO	Teaching Learning-Based Algorithm
LCA	League Championship Algorithm	ACO	Ant Colony Optimization
HBSO	Honey Bee Swarm Optimization	MO	Monkey Optimization

Algorithms inspired by human behaviour are methodologies based solely on how humans act and make decisions. Each individual possesses unique and particular ways of performing activities, which can influence their efficiency. This idea has inspired researchers to create algorithms aimed at solving practical, real-world challenges. Notable examples within this category include the Teaching-Learning-Based Optimization (TLBO) and the League Championship Algorithm (LCA) [20].

Swarm intelligence algorithms derive their principles from the social behaviours observed in insects, animals, fish, and birds. For instance, Kennedy and Eberhart developed the Particle Swarm Optimization (PSO) technique, which mimics how a flock of birds manoeuvrers through a search space to locate the optimal spot. Other examples include Ant Colony Optimization (ACO), Honey Bee Swarm Optimization (HBSO), and Monkey Optimization (MO) [21].

The Smell Agent Optimization (SAO) algorithm is a recent metaheuristic inspired by the sense of smell and the trail-following behaviour exhibited by agents such as humans, dogs, sharks, insects, and bacteria to locate scent sources [23]. SAO has been effectively applied to various problem domains, such as path planning [24], capacitated vehicle routing [25], frequency stabilization in micro-grids [26], hybrid renewable energy systems [23], and coordination schemes in power distribution networks [27]. Like other

metaheuristics, SAO faces challenges related to balancing exploration and exploitation, susceptibility to becoming trapped in local minima, and slow convergence rates.

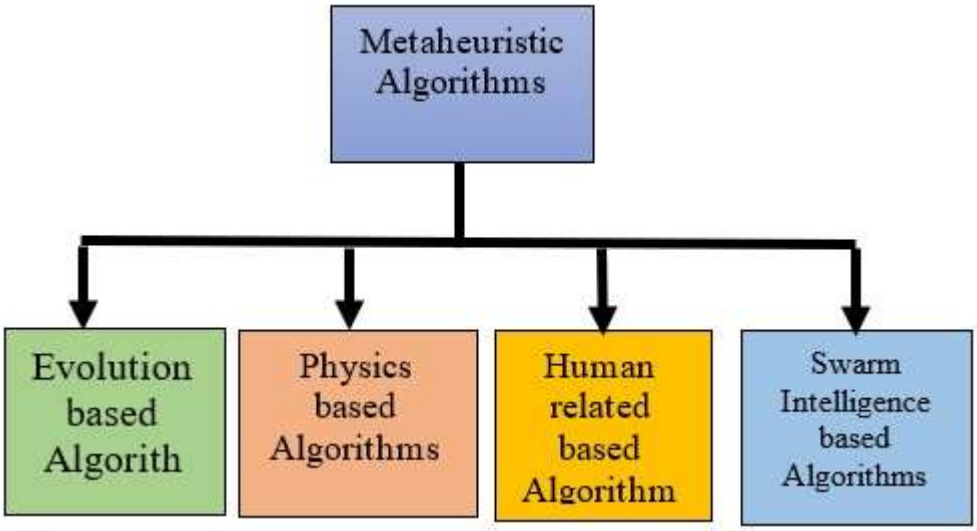

Fig. 1. Categorization of Metaheuristic Algorithms [21]

The rest of the paper is structured as follows. Section 2 discusses the core concepts of the Marine Predator Algorithm and details how the Lévy flight mechanism has been integrated. Section 3 outlines the development of the improved Marine Predator Algorithm (eMPA) and shares the results of its computational testing. Finally, Section 4 assesses the effectiveness of eMPA by comparing it with the standard MPA and various other optimization techniques through a series of experimental analyses.

2.REVIEW OF METAHEURISTIC ALGORITHM

This section focuses on the review of the metaheuristic algorithms such as Particle Swarm Optimization and the Marine Predator Optimization Algorithm as well as the levy flight utilized in this work.

2.1. Particle swarm optimization

Particle Swarm Optimization (PSO), a population-based algorithm inspired by social behaviors observed in nature such as fish schooling and bird flocking, was introduced by Kennedy and Eberhart in 1995 [28]. The method begins by randomly generating a group of particles, mirroring the initialization step common in other evolutionary algorithms like genetic algorithms (GAs). Each particle keeps track of its own best experience, known as the personal best (pbest), along with the best position found by the entire swarm, called the global best (gbest). Throughout the optimization process, particles iteratively update their velocities and positions based on both their personal findings and the swarm's collective knowledge. This dynamic process allows the particles to thoroughly explore the search space and gradually move toward optimal solutions. The updates for particle velocity and position are governed by two key equations [29], with each particle operating within a search space of D dimensions.

Position: $s_i = (s_{i1}, s_{i2}, \ldots\ldots\ldots\ldots s_{ik})$

Velocity: $v_i = (v_{i1}, v_{i2}, \ldots\ldots\ldots\ldots\ldots v_{ik})$

At each iteration, the velocity and position are updated as follows:

Velocity update Equation:

$$V_i^{k+1} = wV_i^k + c_1 * rand_1\left(P_{best} - S_i^k\right) + c_2 * rand_2\left(G_{best} - S_i^k\right) \quad (1)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (2)$$

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}}\right) \times iter \quad (3)$$

Where,

V_i^k is the Velocity of agent i at kth iteration

V_i^{k+1} is the Velocity of agent i at (k+1)th iteration

w is the inertia weight

c_1 and c_2 are the weight factor (0 to 4)

S_i^k is the Current position of agent i at kth iteration

S_i^{k+1} is the Current position of agent i at (k+1)th iteration

$iter_{max}$ is the Maximum iteration number

iter is the Current iteration number

P_{best} is the personal best of agent i

G_{best} is the Global best of the group

w_{max} is the Initial value of the inertia weight

w_{min} is the Final value of the inertia weight

$rand_1$ *and* $rand_2$ are the random numbers usually chosen between [0,1].

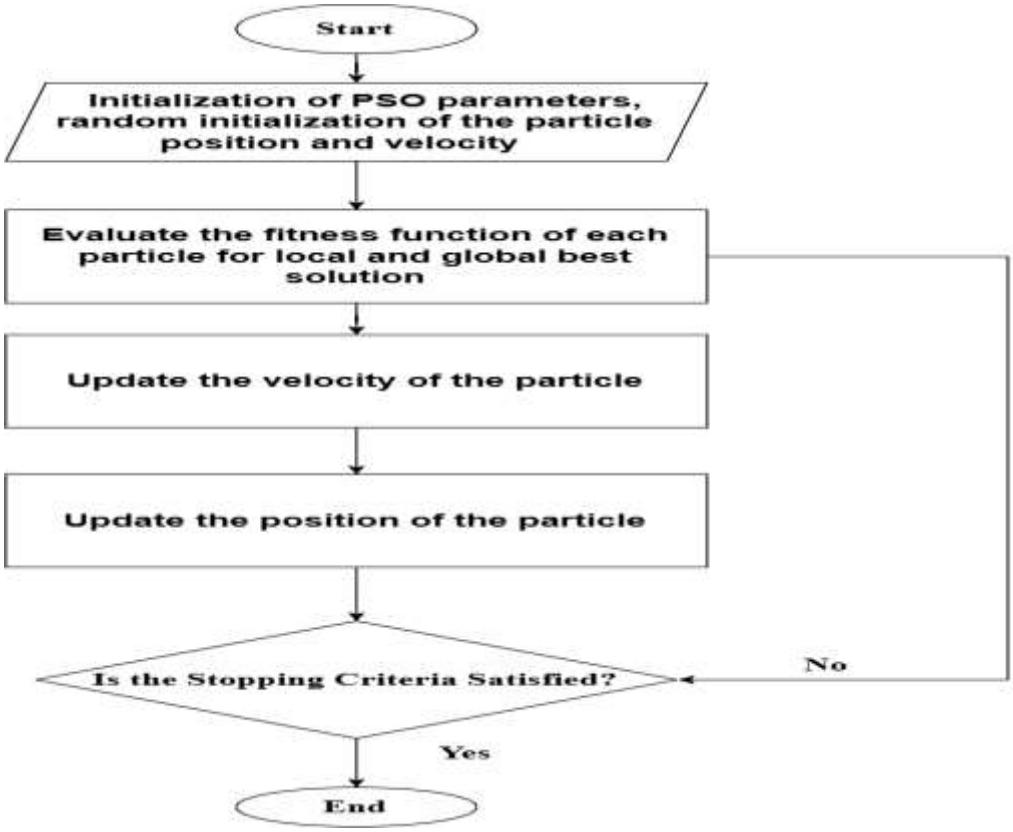

Figure 2: PSO Flowchart

2.2. Marine predator algorithm

The Marine Predator Algorithm (MPA) is a nature-inspired meta-heuristic designed to simulate the hunting strategies and movement behaviors of marine predators such as sharks and dolphins in aquatic environments. It strategically combines exploration (broad search) and exploitation (fine-tuning) phases to effectively discover optimal or near-optimal solutions in complex search spaces [30]. The algorithm mimics predator-prey interactions, adapting these natural behaviors into a computational framework to solve various optimization problems efficiently. The core of MPA mathematical formulation involves the representation of candidate solutions (also called agents or predators) and their evolution over iterations. The main equations can be summarized as:

1.Position Update Equation: Each predator's position in the search space at iteration $t + 1$ is updated based on a blend of exploration and exploitation strategies:

$$x_i^{t+1} = x_i^t + R \times (x^{best} - x_i^t) + S \times Random\ walk \quad (4)$$

where:

x_i^t: position vector of the *i-th* predator at iteration *t*

x^{best}: position vector of the current best solution

R: exploitation factor controlling convergence speed

S: exploration factor balancing diversification

Random Walk: stochastic component representing exploration, often modelled as Gaussian distribution

2.Predation Strategy Transition: The algorithm switches between different hunting strategies based on the predator's energy and prey response:

$$strategy_t = \begin{cases} Exploration \ \ if \ \ E_t > E_{threshold} \\ \\ Exploitation \ \ \ Otherwise \end{cases} \quad (5)$$

where E_t is the predator's energy or motivation, which diminishes or increases based on success in capturing prey.

Manipulation of Parameters: Adaptive adjustment of parameters like R and S over iterations to balance exploration/exploitation [31]:

$$R_t = R_{max} - \left(\frac{R_{max} - R_{min}}{Max_iter} \times t\right) \quad (6)$$

2.3. Levy flight

This is a stochastic random-walk search strategy that has become an important component of modern meta-heuristic optimization algorithms because of its ability to improve global exploration while maintaining effective local exploitation in complex, nonlinear, multimodal, and high-dimensional optimization problems [31]. Unlike traditional Gaussian random walks, Lévy Flight generates many short movements punctuated by infrequent but significant long-distance jumps, which are determined by a heavy-tailed probability distribution. This approach helps search agents overcome local optima and enhances their ability to explore new, unvisited regions of the search space more effectively. This strategy has been successfully integrated into various optimization techniques such as Particle Swarm Optimization (PSO), Arithmetic Optimization Algorithm (AOA), Grey Wolf Optimizer (GWO), and Prairie Dog Optimization (PDO), aiming to improve convergence speed, maintain diversity in the search process, increase robustness, and boost solution precision [32]. The mathematical basis starts with the position update rule, denoted as equation (7):

$$x_i^{t+1} = x_i^t + \alpha \otimes Levy(\beta) \quad (7)$$

Where x_i^t and x_i^{t+1} denote the current and updated position of i^{th} search agent, α is the step size scaling factor, and $Levy(\beta)$ represents the Lévy-distributed random step. The Lévy probability distribution is given by equation (8)

$$P(s) \propto |s|^{-1-\beta} \quad (8)$$

$1 < \beta < 2$ and s is the random step length, indicating that short movements occur frequently while long jumps occur with lower probability but substantially improve global exploration. To generate the Lévy step, Mantegna's algorithm is commonly employed and is expressed as equation (9)

$$Levy(\beta) = \frac{\mu}{|v|^{\frac{1}{\beta}}} \quad (9)$$

where $\mu \sim N(0, {\sigma_u}^2)$ and $v \sim N(0,1)$. The corresponding standard deviation is determined using equation (10):

$$\sigma_u = \left[\frac{\Gamma(1+\beta)sin\left(\frac{\pi\beta}{2}\right)}{\Gamma((1+\beta)/2)\beta 2^{\frac{(\beta-1)}{2}}}\right]^{\frac{1}{\beta}} \quad (10)$$

Here, Γ (·) represents the Gamma function. Together, these mathematical models offer an effective search strategy that markedly boosts optimization results by maintaining a balance between exploration and exploitation. This approach helps prevent early convergence and leads to higher-quality solutions across various engineering fields, including renewable energy optimization, tuning controller parameters, and improving power system stability [34].

3. Development of an Enhanced Marine Predator Algorithm (eMPA)

This section provides a detailed overview of the development process for the Levy Flight MPA and describes the benchmark functions employed to assess the performance of the optimization algorithms.

3.1. Enhanced Marine Predator Algorithm (eMPA)

The eMPA integrates the concept of Levy flight into the original MPA to enhance the algorithm's exploration capabilities without compromising its exploitation skills. The subsequent subsection presents the detailed mathematical formulations for each modified mode that incorporates Levy flight. The hybrid approach of combining the original MPA with Levy flight aims to strengthen the algorithm's exploitation ability while maintaining effective exploration. As outlined in the problem statement, the original MPA simulates the hunting strategies of marine predators and employs probabilistic steps based on Levy distribution to facilitate a robust global search. The detailed mathematical expressions for each modified mode utilizing Levy flight are provided in the following section.

1. Parameter Initialization

The following parameters will be utilized:

$N_p = 50$: Number of predators

D=2: Dimensionality of the search space

G=100: Maximum number of generations

$\alpha =$ 0.5: Probability of global search (Levy-driven)

β, γ, δ = 0.5: Control parameters for local and random walks

$\in =$ 0.01: Levy weight for local exploitation

$[L, U] = [100, 100]$: Lower and upper bounds of the search space

2. Position Initialization

Each predator's position is initialized randomly within the search space:

$$x_i = \mathrm{L} + (U - L).rand(D), i = 1,2, \ldots N_p \quad (11)$$

3. Optimization Loop

For each generation *g* = 1 to *G*, the following steps are executed:

i. Fitness Evaluation: Each predator' s position is evaluated using a selected benchmark function $f(\cdot)$

ii. Leader Identification: The predator with the minimum fitness is selected as the leader:

$$x_{leader} = \arg\, min_i f(x_i) \quad (12)$$

iii. Position Update: Each predator updates its position based on one of three behaviors:

a. Global Exploration (with probability α)

$$x_i = x_{leader} + \beta . Levy(1, D) \quad (13)$$

b. Local Exploitation (with probability γ):

$$x_i = x_i + \delta . (x_j - x_i) + \epsilon . Levy(1, D) \quad (14)$$

where x_j is a randomly selected predator.

c. Random Reinitialization:

$$x_i = \mathrm{L} + (U - L) . rand(1, D) \quad (15)$$

4.Boundary Handling: Updated positions are clamped to the boundary:

$$x_i = \max(min(x_i, U), L) \quad (16)$$

4. Progress Logging: The best fitness value is stored for convergence analysis and the benchmark functions are in Table 1

Levy Flight Function

Levy flight is implemented as:

$$Levy(n, d) = \frac{u}{\lceil v \rceil^{\frac{1}{\beta}}} \ , u \sim N(0, \sigma^2), v \sim N(0, 1)$$

where β = 1.5 and σ is computed as:

$$\sigma = \left(\frac{\Gamma(1+\beta) . sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{(1+\beta)}{2}\right) . \beta . 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}}$$

Convergence visualization

The algorithm plots the best fitness per generation to visualize convergence:

$$plot(g, f(x_{leader}))$$

The flowchart of the proposed algorithm is showing Figure 3

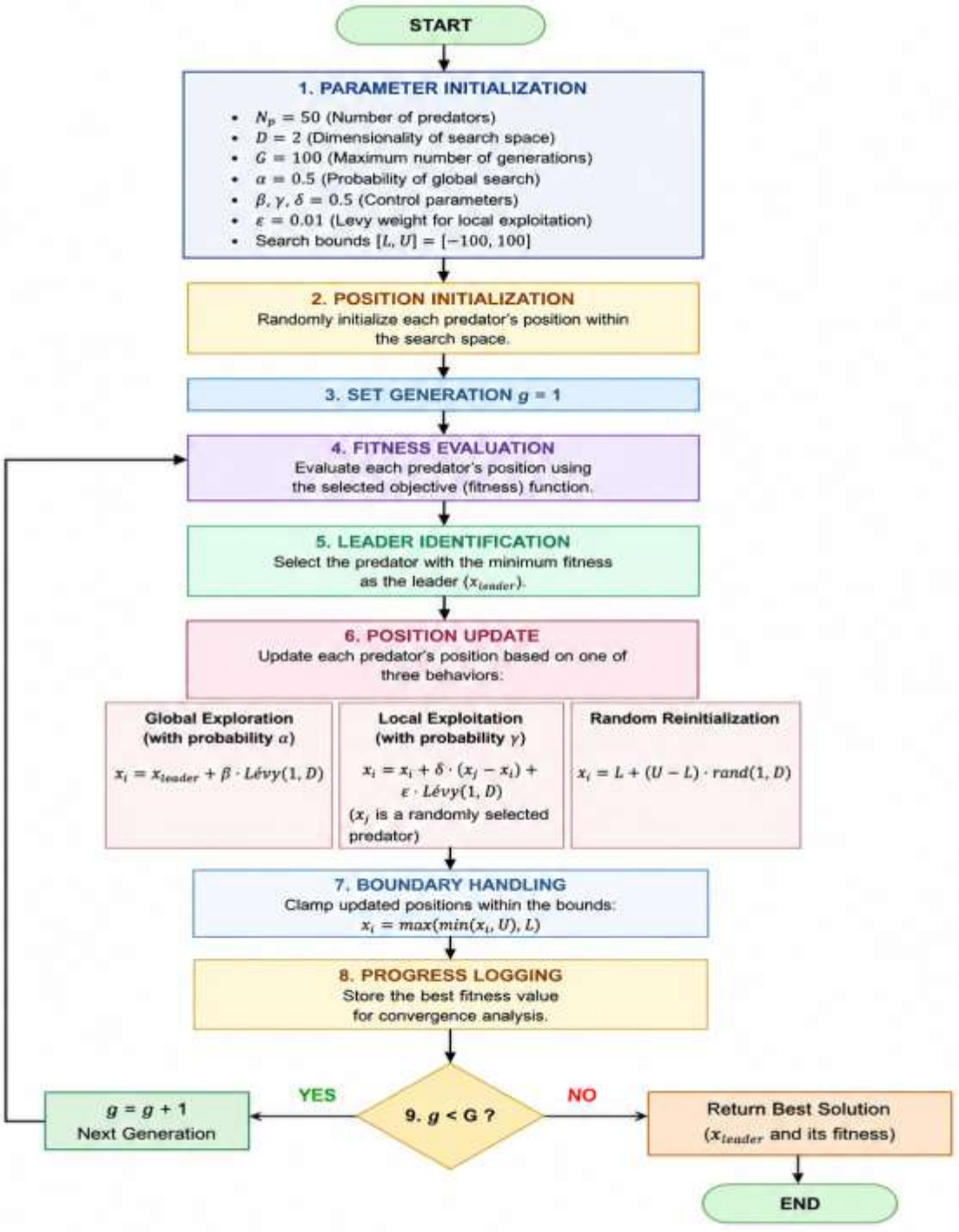

Figure 3: Flowchart of eMPA

3.3.2.1 Evaluation of eMPA on Benchmark Functions

The performance of an optimization algorithm cannot be accurately assessed by the specific problems it addresses, especially if those problems are highly specialized and lack diverse characteristics. To thoroughly evaluate the proposed algorithm's effectiveness, a selection of seven benchmark functions will

be used, chosen according to the attributes outlined in Table 1. The complexity of each of these functions is represented through a corresponding metric function.

Table 1:Benchmark Function

Fn	Names	Description
F_1	Adjiman	$f(x_1,x_2)=\cos(x_1)\sin(x_2)-\frac{x_1}{x_2^2+1}$
F_2	Bohachevsky1	$f(x_1,x_2)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$
F_3	Michalewicz	$f(x)=-\sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$
F_4	Schaffer	$f(x)=\sum_{i=1}^{30}\left(x_i^2+x_{i+1}^2\right)^{0.5}\left\{\left[\sin 50\left(x_i^2+x_{i+1}^2\right)^{0.1}\right]^2\right\}$
F_5	Styblinki's Tang	$f(x)=\sum_{i=1}^{n}(x_i^4-16x_i^2+5x_i)$
F_6	Box-Betts	$f(x_1,x_2)=\sum_{i=1}^{k}(e^{-0.1(i+1)x_1}-e^{-0.1(i+1)x_2}-[(e^{-0.1(i+1)})-e^{-(i+1)x_3})])^2$
F_7	Michalewicz 5	$f(x)=-\sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$
F_8	Miele Cantrell	$f(X)=(e^{-x_1}-e^{-x_2})^4+100(x_2-x_3)^6+(\tan(x_3-x_4))^4+x_1^8$
F_9	Step	$f(x)=\sum_{i=1}^{n}(\lvert x_i\rvert+0.5)^2$
F_{10}	Michalewicz 10	$f(x_1,x_2)=(4-2.1x_1^2+\frac{x_1^4}{3})x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$
F_{11}	Brown	$f(\mathrm{X})=\sum_{i=1}^{n-1}(x_i^2)^{(x_{i+1}^2+1)}+(x_{i+1}^2)^{(x_i^2+1)}$
F_{12}	Ellipsoid	$f(x)=\sum_{i=1}^{n}i^2x_i^2$
F_{13}	Quartic	$f(X)=\sum_{i=1}^{D}ix_i^4+random[0,1]$
F_{14}	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}((x_i-1)^2+100(x_{i+1}-x_i^2)^2)$
F_{15}	Salomon	$f(X)=1-\cos(2\pi\sqrt{\sum_{i=1}^{n}x_i^2})+0.1\sqrt{\sum_{i=1}^{n}x_i^2}$

4. RESULTS AND DISCUSSION

This subsection outlines the experimental findings and provides an analysis of the results. Additionally, it includes an examination of the convergence behaviour observed in the experiments.

4.1 The Performance of the Algorithms on Benchmark Functions

Tables 2 through 4 display the benchmark outcomes obtained from four optimization techniques: the improved Marine Predator Algorithm with Levy flight (eMPA), the standard Marine Predator Algorithm (MPA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO) across various dimensions. Table 2 shows results for two-dimensional benchmark functions, while Table 3 presents data for moderate-dimensional functions (3D, 4D, 5D, and 10D). Table 4 covers results for higher-dimensional benchmark functions (30D). The proposed eMPA was tested alongside MPA, PSO, and GWO, with each algorithm run independently 10 times, each with 300 iterations. For each, the best, worst, average, standard deviation, and ranking were recorded. The optimal result is highlighted in bold. Rankings were primarily based on the best objective function value achieved; when multiple algorithms shared the same best value, the average, standard deviation, and worst-case results were used to compare their performance. Algorithms with identical performance received the same rank, and subsequent ranks were adjusted accordingly. If two algorithms are both rank r, the next algorithm is rank r+2 and so on. Finally, the average rank across all benchmark functions was calculated to determine the overall performance and final ranking of the algorithms.

Table 2: Performance of the Algorithm on Bidimensional Benchmark Functions

Functions	Performance Matrix	eMPA	MPA	PSO	GWO	Global Best
F1 Admijan	Mean	-2.0218	-2.0218	-2.02180	-2.02180	-2.0218
	Best	**-2.0218**	**-2.0218**	**-2.02180**	**-2.02180**	
	Std	2.146e-08	4.801e-08	4.6811e-16	3.039e-12	
	Worst	-2.0218	-2.0218	-2.02180	-2.02180	
	Rank	3	4	1	2	
F2 Bohachevsky	Mean	0.0	0.0	0.0	0.0	0
	Best	**0.0**	**0.0**	**0.0**	**0.0**	
	Std	0.0	0.0	0.0	0.0	
	Worst	0.0	0.0	0.0	0.0	
	Rank	1	1	1	1	
F3 Michalewicz2	Mean	-1.997	-1.999	-1.999	-1.999	-1.8013
	Best	-1.996	-1.999	-1.999	-1.999	
	Std	6.472e-05	5.965e-05	8.724e-05	6.652e-05	
	Worst	-1.999	-1.999	-1.9999	-1.999	
	Rank	1	2	4	3	
F4 Schaffer						0
	Mean	0.000	0.000	2.719	7.746	

				e-30	e-11	
	Best	**0.000**	**0.000**	**0.000**	**0.000**	
	Std	0.000	0.000	8.5987 e-30	2.252 e-10	
	Worst	0.000	0.000	2.7191e-29	7.165e-10	
	Rank	1	1	3	4	
F5 StyblinskiTang	Mean	-78.332	-78.332	-78.332	-78.332	**-78.332**
	Best	**-78.332**	**-78.332**	**-78.332**	**-78.332**	
	Std	2.650e-06	2.800e-06	1.4979e-06	8.851e-06	
	Worst	-78.332	-78.332	-78.332	-78.332	
	Rank	2	3	1	4	
	Count of Global optimal	4	4	4	4	
	Avg Ranking	1.4	2.75	2.0	2.8	
	Final Ranking	**1**	**3**	**2**	**4**	

The data presented in Table 2 illustrate the performance of eMPA, MPA, PSO, and GWO across five two-dimensional benchmark functions: F1, F2, F3, F4, and F5. These algorithms were assessed using performance indicators such as the mean, best, standard deviation, worst value, and a ranked order derived from their optimization outcomes. The results indicate that all four algorithms successfully found the global optimal solution in four out of the five functions (F1, F2, F4, and F5), accounting for 80% of the tested bidimensional benchmarks. For the F3 function, none of the algorithms achieved the true global optimum; however, the solution provided by eMPA was marginally better than those found by MPA, PSO, and GWO. This outcome suggests that the algorithms, especially in comparison to eMPA, are standard methods regularly employed in solving optimization challenges. Regarding average rankings, eMPA recorded the lowest mean rank of 1.40, followed by PSO with 2.00, MPA with 2.75, and GWO with 2.80. Therefore, eMPA secured the top overall position, with PSO, MPA, and GWO trailing behind. These findings highlight the improved performance of eMPA, likely due to the enhancements that foster a better balance between exploration and exploitation, enabling it to outperform the other methods. It is also notable that even though all algorithms reached the same number of global optima, eMPA still achieved the best final ranking. This indicates that the total count of global optima found alone does not determine an algorithm's overall superiority; other factors such as the mean solution quality, worst-case results, and standard deviation also play crucial roles in evaluating overall performance.

Table 3: Performance of the Algorithm on Moderate-dimensional Benchmark Functions

Functions	Performance Matrix	eMPA	MPA	PSO	GWO	Global Best
F6 Box-Betts	Mean	0.000	0.000	0.000	0.000	0
	Best	**0.000**	**0.000**	**0.000**	**0.000**	
	Std	0.000	0.000	0.000	0.000	
	Worst	0.000	0.000	0.000	0.000	
	Rank	1	1	1	1	
F7 Michalewicz5	Mean	-4.994	-4.996	-5.00	-4.999	-4.6877
	Best	-4.996	-4.998	-5.00	-4.999	
	Std	2.6096 e-04	3.152e-06	0.00	1.3680 e-04	
	Worst	-4.991	-4.999	-5.00	-4.999	
	Rank	1	2	4	3	
F8 Miele Cantrell	Mean	0.000	0.000	0.000	0.000	0
	Best	**0.000**	**0.000**	**0.000**	**0.000**	
	Std	0.000	0.000	0.000	0.000	
	Worst	0.000	0.000	0.000	0.000	
	Rank	1	1	1	1	
F9	Mean	0.000	0.000	0.000	0.000	0
	Best	**0.000**	**0.000**	**0.000**	**0.000**	
	Std	0.000	0.000	0.000	0.000	
	Worst	0.000	0.000	0.000	0.000	
	Rank	1	1	1	1	
F10 Michalewicz10	Mean	-9.998	-9.998	-10.00	-9.998	-9.6602
	Best	-9.997	-9.997	-10.00	-9.998	
	Std	3.2823 e-04	6.1689 e-04	0.00	3.5509e-04	
	Worst	-9.999	-9.999	-10.00	-9.999	
	Rank	2	1	4	3	
	Count of Global optimal	3	3	3	3	
	Avg Ranking	1.2	1.2	2.2	1.8	
	Final Ranking	**1**	**1**	**4**	**3**	

Similarly, the data in Table 3 for functions with moderate dimensions reveal that eMPA, MPA, PSO, and GWO each achieved the global optimum in three out of five cases (F6, F8, and F9), accounting for 60% of these functions. In the cases of F7 and F10, none of the algorithms reached the absolute best solution; however, eMPA demonstrated results that were marginally better than those of the other three methods for F7. For F10, MPA performed slightly better than the remaining algorithms, although none attained the optimal value. The average rankings reinforce this trend, with both eMPA and MPA sharing the top position with an average rank of 1.2, while GWO and PSO received average ranks of 1.8 and 2.2, respectively. From an optimization standpoint, these outcomes suggest that eMPA maintains a balanced approach between exploration and exploitation. Its ability to secure the best ranking on F7 and hold competitive standings on F10 indicates that the enhancements integrated into eMPA improve its search efficiency without hindering convergence. This balance is especially vital for problems with multiple local optima, where too much exploitation can lead to premature convergence, and excessive exploration may slow progress.

Table 4: Performance of the Algorithm on Multi-dimensional Benchmark Functions

Functions	**Performance Metrics**	**eMPA**	**MPA**	**PSO**	**GWO**	**Global Best**
F11 Brown	Mean	0.000	0.000	0.000	0.000	0
	Best	**0.000**	**0.000**	**0.000**	**0.000**	
	Std	0.000	0.000	0.000	0.000	
	Worst	0.000	0.000	0.000	0.000	
	Rank	1	1	1	1	
F12 Ellipsoid	Mean	0. 0	0. 0	0. 0	0. 0	0
	Best	**0. 0**	**0. 0**	**0. 0**	**0. 0**	
	Std	0. 0	0. 0	0. 0	0. 0	
	Worst	0. 0	0. 0	0. 0	0. 0	
	Rank	1	1	1	1	
F13 Quartic	Mean	1.987	2.388	7.479e-04	6.489e-04	0
	Best	0.635	1.625	3.321e-06	7.431e-05	
	Std	0.293	0.4987	4.667e-03	5.617e-04	
	Worst	2.6410	3.3625	6.325e-03	9.432e-04	
	Rank	3	4	1	2	
F14 Rosenbrock	Mean	4.562e-09	3.512e-08	4.314e-06	2.329e-06	0
	Best	**2.538e-13**	3.028e-13	4.206e-13	4.161e-06	
	Std	2.673e-08	8.615e-07	2.321e-06	2.321e-06	
	Worst	3.274e-08	4.254e-05	7.124e-06	7.124e-06	
	Rank	1	2	4	3	
F15 Salomon	Mean	0. 0	0. 0	0. 0	0. 0	0
	Best	**0. 0**	**0. 0**	**0. 0**	**0. 0**	
	Std	0. 0	0. 0	0. 0	0. 0	
	Worst	0. 0	0. 0	0. 0	0. 0	
	Rank	1	1	1	1	
	Count of Global optimal	4	4	4	4	
	Avg Ranking	1.4	1.8	1.6	1.6	
	Final Ranking	**1**	**4**	**2**	**2**	

Also, the results obtained in Table 4 for the Multimodal-dimensional functions also shows that eMPA, MPA, PSO, and GWO obtained the global best in 4 (F11, F12, F14 and F15) out of 5 which translate to 80% of the multimodal-dimensional functions. In F13 none of the algorithms obtained the best result while PSO provide much better than the other three algorithms. This performance clearly explained the no free lunch theory, that no optimization algorithm is of general application. In terms of ranking, eMPA, achieves the best overall ranking of 1 among the four algorithms. Although all four algorithms obtained the global optimum on 4 out of the 5 test functions, indicating a 80% global-optimal success rate, their overall performance differs when the mean, best, standard deviation, and worst results are considered together. eMPA obtained the lowest average ranking of 1.4, compared with 1.8 for MPA, 1.6 for PSO, and 1.6 for GWO. Therefore, based on the final ranking, eMPA is the best-performing algorithm, followed by PSO and GWO in joint second position, while MPA ranks fourth. Overall, the results demonstrate that eMPA provides the most consistent and competitive optimization performance across the tested benchmark functions. Importantly, the superiority of eMPA is not based solely on obtaining the global optimum, since all algorithms achieved it on the four functions; rather, it is demonstrated by its overall ranking and superior numerical performance on the more challenging multimodal benchmark functions.

4.2 Convergence analysis

The convergence plot for the unimodal functions is as show in Fig 4. to Fig 8. An algorithm whose curve decreases rapidly in the early iterations demonstrates faster convergence and a strong exploitation capability, whereas a slower but steady decline indicates a more gradual search process. The algorithm that reaches the lowest objective function value and maintains this value with minimal subsequent variation can be considered to have achieved the best optimization performance.

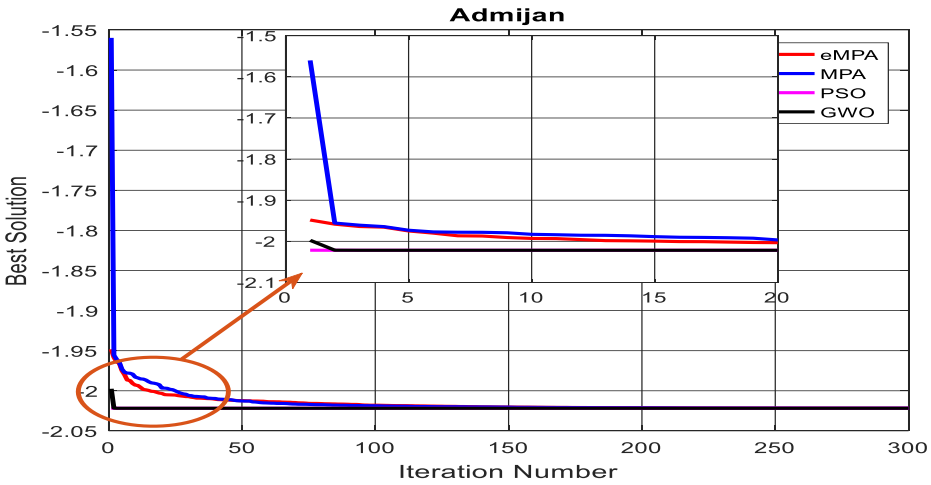

Fig 4: Admijan Function (F1)

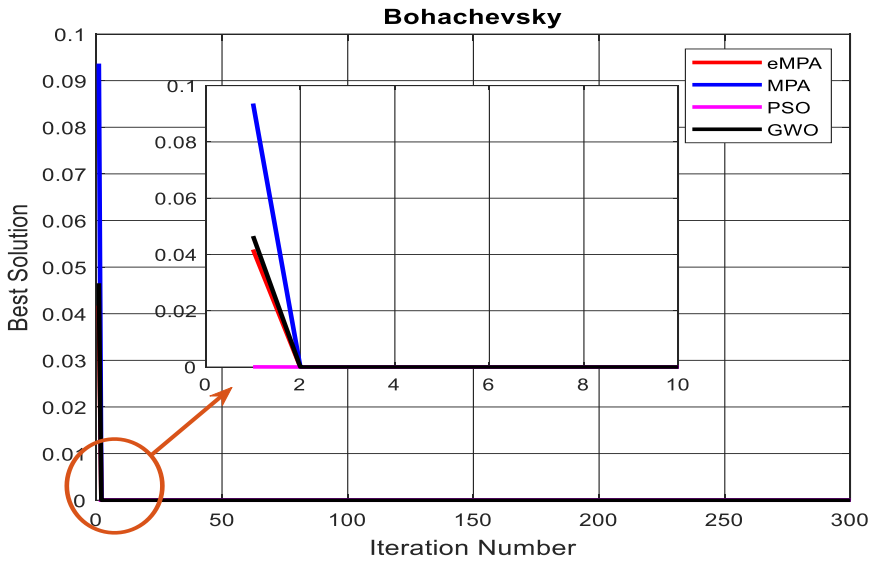

Fig 5: AdmijanFunction (F2)

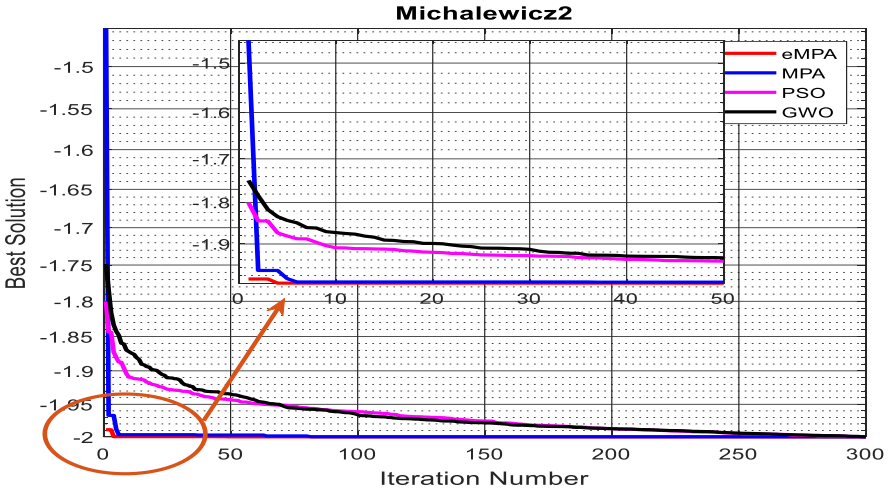

Fig 6: Michalewicz2 Function (F3)

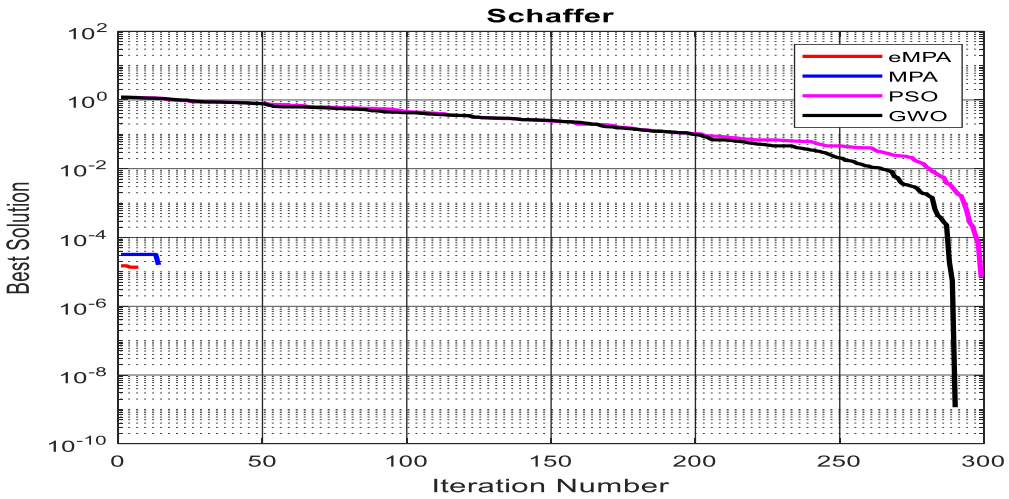

Fig 7: Schaffer Function (F4)

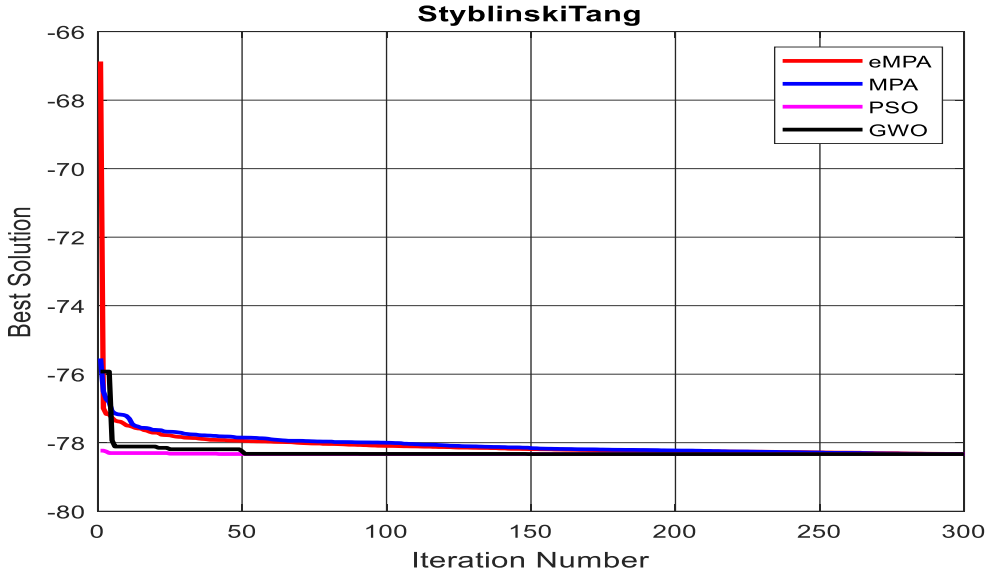

Fig 8: StyblinskiTang Function (F5)

From the convergence result of the algorithms presented from Fig 4 to Fig 8 it is observed eMPA converges faster in 3 functions (F2, F3 and F4) out of 5 which is 60% of the functions. In F1, GWO converges faster than other three and in F5, PSO offers the fastest convergence. Although PSO convergence faster F5, eMPA continues to search in other to obtain a better solution. The over all performance shows the superior convergence of eMPA over the other algorithms whivh is attributed to the levyflight introduced in the MPA.

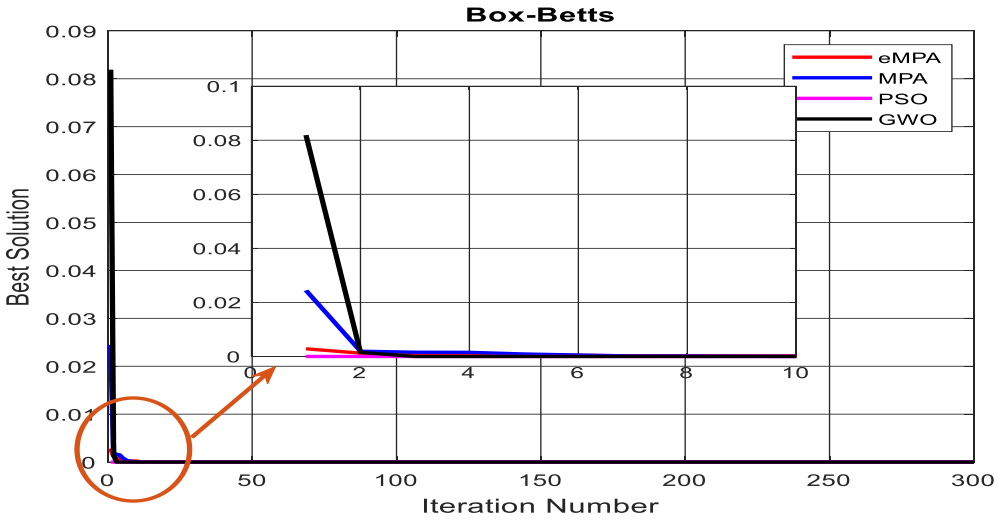

Fig 9: Box-Belt Function (F6)

Fig 10: Michalewicz5 Function (P 7)

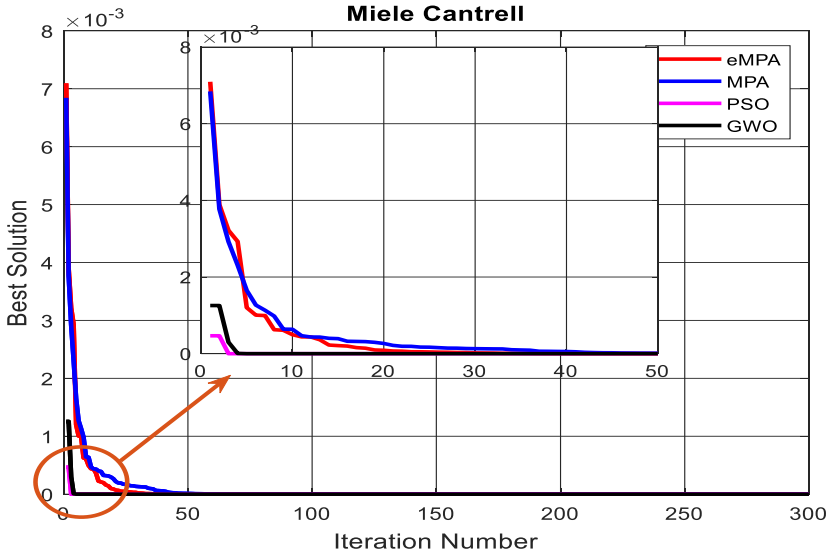

Fig 11: Miele Cantrell Function (F8)

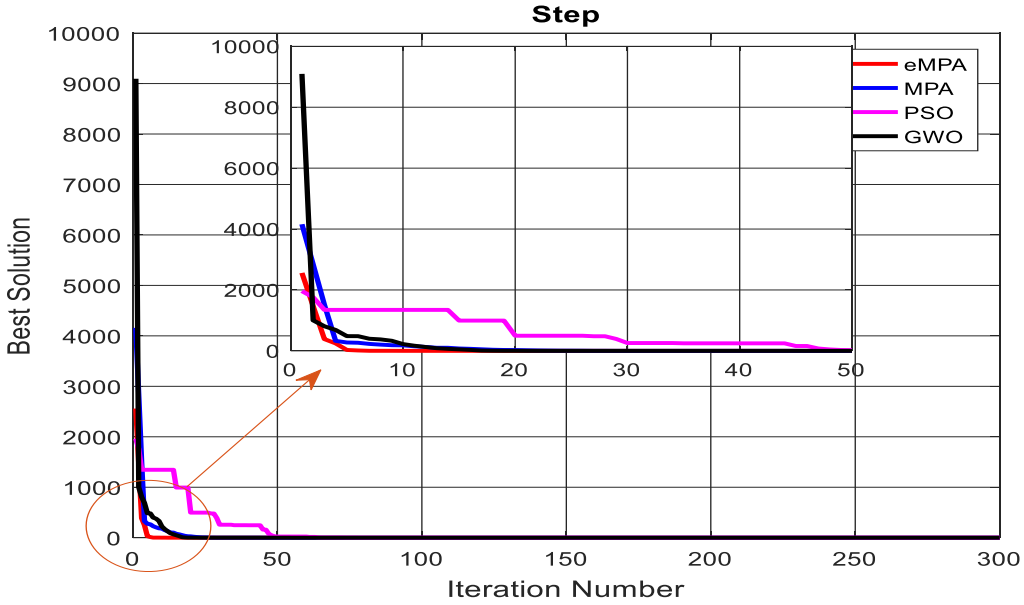

Fig 12: Step Function (F9)

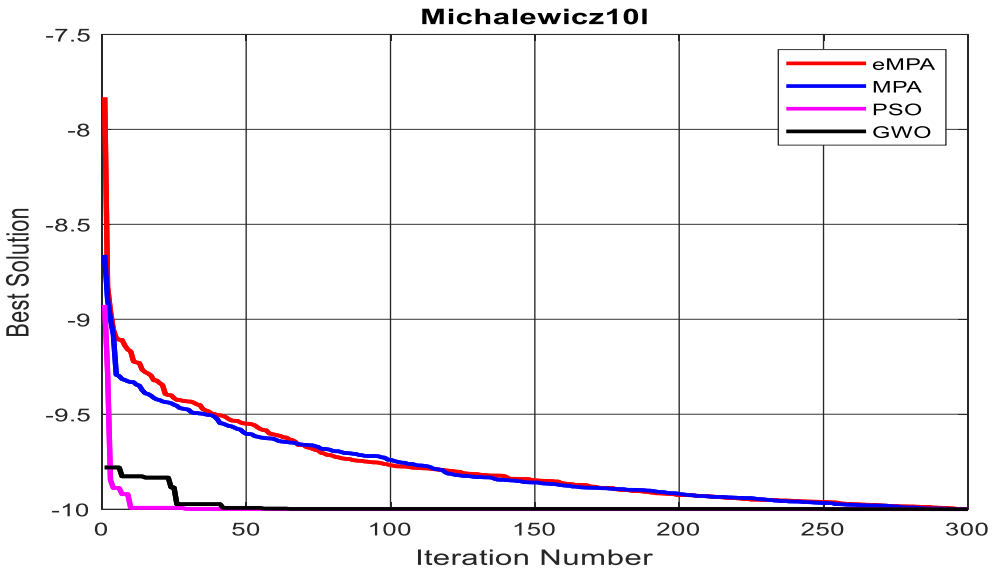

Fig 13: Michalewicz10 Function (F10)

Similarly, the convergence result of the algorithms presented from Fig 9 to Fig 13, it is observed that eMPA converges faster in 2 functions (F6 and F9) out of 5 which is 40% of the functions, MPA converges in 1 function (F6) which is 20%, PSO converges faster in 3 functions (F6, F8 and F10) which is 60% and GWO converges in 2 functions (F1 and F7) which is 40% of the moderate-dimensional functions. The superior performance of PSO over eMPA and the other algorithms is attibuted to its high exploitation ability over exploration. These results demonstrate the importance of maintaining an appropriate exploration–exploitation balance: excessively rapid convergence may lead to premature convergence and loss of population diversity, whereas controlled convergence can provide better global-search capability.

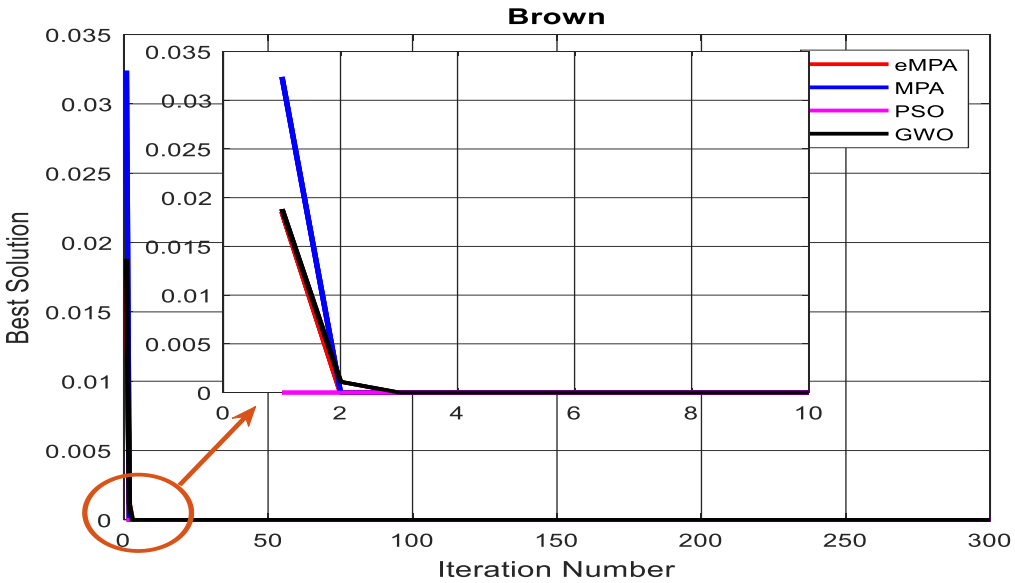

Fig 14: Brow Function (F11)

Fig 15: Ellipsoid Function (F12)

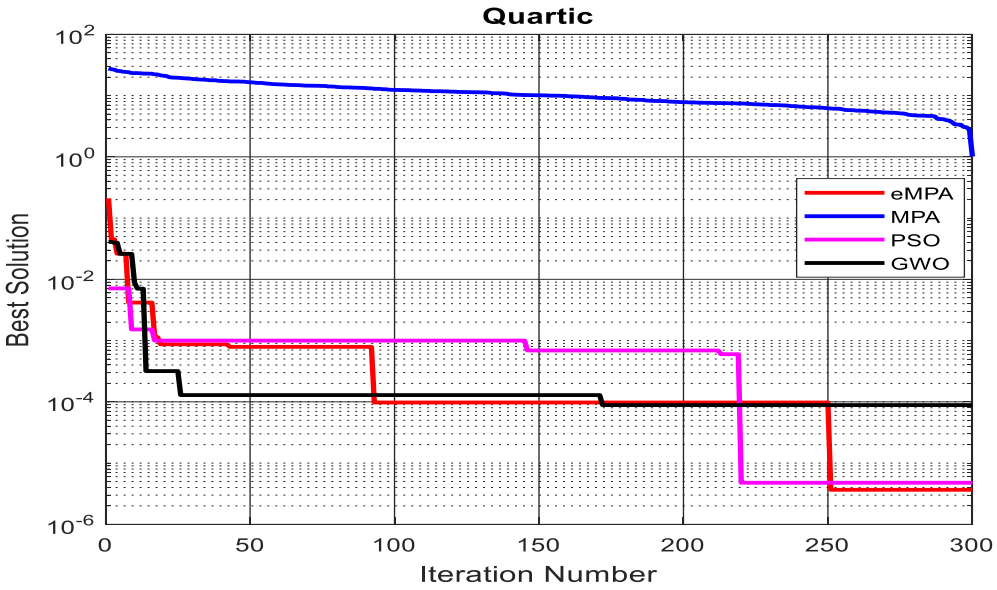

Fig 16: Quarticdcecece Function (F13)

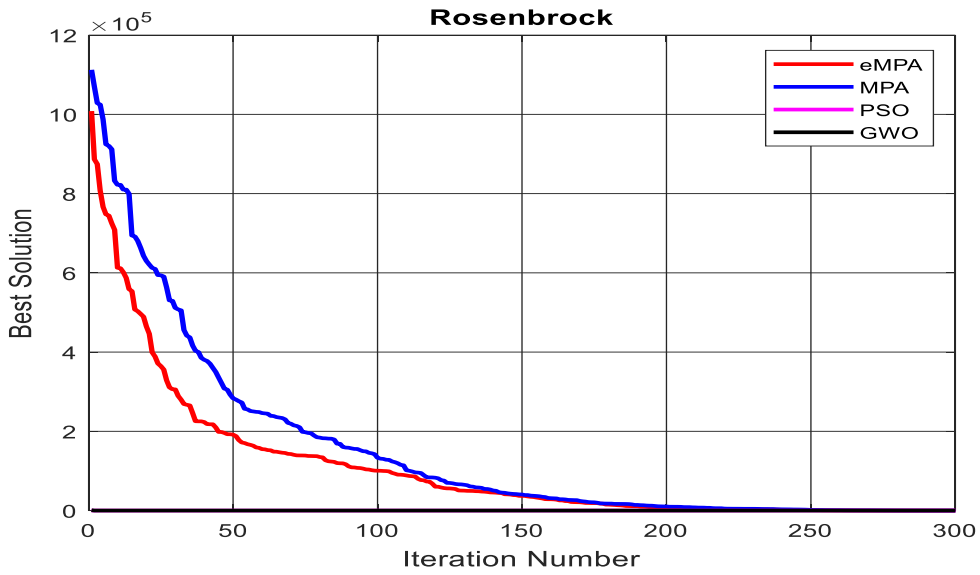

Fig 17: Rosenbrock Function (F14)

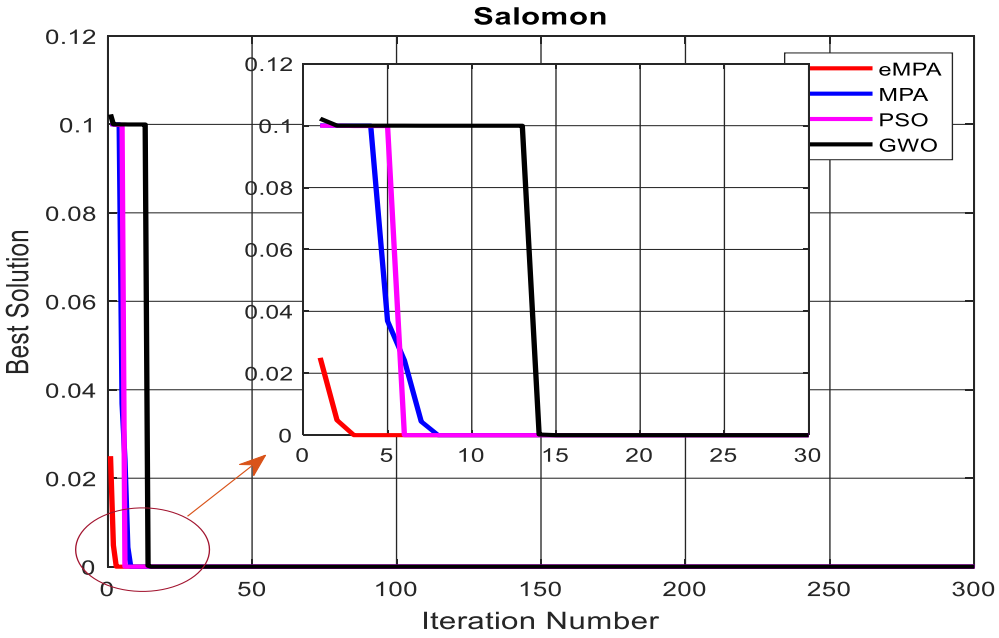

Fig 18: Salon Function (F15)

Finally, the convergence plot for the multimodal functions is also presented from Fig 14 to Fig 18. From these plots it is observed that eMPA converges faster in 3 functions (F11, F12 and F15) out of 5 which is 60% of the functions, MPA converges in 2 function (F11 and F12) which is 40%, PSO converges faster in 2 functions (F11 and F13) which is 20% of the functios and GWO converges in 1 functions (F11 and F12) which is 40% of the multimodal-dimensional functions. In F14 the high exploitation ability of PSO and GWO shows that both are in a straight horizontal line while eMPA and MPA takes longer number of iterations to converge due to high level of exploration over exploitation. These result also shows the superior performance of eMPA over the other algorithms.

5. CONCLUSION

This study proposed a novel metaheuristic optimization technique termed enhance Marine Predator Algorithm (eMPA), which incorporates Levy Flight into the conventional Marine Predator Algorithm (MPA). The integration of Levy Flight into this mode enhances the search mechanism by improving population diversity and strengthening the overall optimization performance. The effectiveness of eMPA was assessed using fifteen benchmark functions with different characteristics and complexities. The statistical analysis demonstrated that the results demonstrate that the enhanced Marine Predator Algorithm (eMPA) consistently outperformed or remained highly competitive with MPA, PSO, and GWO across the tested benchmark functions. For the bidimensional functions, all four algorithms reached the global optimum in four out of five functions (80%), but eMPA achieved the best overall ranking with an average rank of 1.40, followed by PSO (2.00), MPA (2.75), and GWO (2.80). For the moderate-dimensional functions, eMPA and MPA jointly obtained the best average rank of 1.20, with eMPA showing superior performance on F7 and competitive results on F10. Similarly, for the multimodal functions, the algorithms achieved the global optimum in four out of five cases (80%); however, eMPA recorded the best overall ranking of 1.0, supported by its strong mean, best, standard deviation, and worst-case results. Overall, the findings indicate that the modifications incorporated into eMPA improved the balance between exploration and exploitation, enhanced search capability and convergence, and provided more consistent optimization performance, particularly for complex multimodal problems. Also, in terms of convergence, the simulation result shows eMPA converges faster in 3 functions (F2, F3 and F4) out of 5 which is 60% of the functions over PSO and MPA and GWO. Also, in terms of moderate dimensional benchmark functions, it is observed that eMPA converges faster in 2 functions (F6 and F9) out of 5 which is 40% of the functions, MPA converges in function (F6) which is 20%, PSO converges faster in 3 functions (F6, F8 and F10) which is 60% and GWO converges in 2 functions (F1 and F7) which is 40% of the moderate-dimensional functions. The superior performance of PSO over eMPA and the other algorithms is attibuted to its high exploitation ability over exploration. These results demonstrate the importance of maintaining an appropriate exploration–exploitation balance: excessively rapid convergence may lead to premature convergence and loss of population diversity, whereas controlled convergence can provide better global-search capability. Finally, in terms of multimodal functions is also presented from Fig 14 to Fig 18. From these plots it is observed that eMPA converges faster in 3 functions (F11, F12 and F15) out of 5 which is 60% of the functions, MPA converges in 2 function (F11 and F12) which is 40%, PSO converges faster in 2 functions (F11 and F13) which is 20% of the functios and GWO converges in 1 functions (F11 and F12) which is 40% of the multimodal-dimensional functions. In F14 the high exploitation ability of PSO and GWO shows that both are in a straight horizontal line while eMPA and MPA takes longer number of iterations to converge due to high level of exploration over exploitation. These result also shows the superior performance of eMPA over the other algorithms.

REFERENCES

[1] Yang, X.-S. “Nature-inspired metaheuristic algorithms: Success and new challenges”. 2012. p1211.6658.

[2] Yılmaz, S., & Küçüksille, E. U. “A new modification approach on bat algorithm for solving optimization problems”. Applied Soft Computing, 2015 p28, 259-275.

[3] Yang, X.-S. “Firefly algorithm, stochastic test functions and design optimization”. 2010a. International Journal of Bio-Inspired Computation, 2(2), 78-84.

[4] Shehab, M., Khader, A. T., & Al-Betar, M. A. “A survey on applications and variants of the cuckoo search algorithm”. 2017. Applied Soft Computing.

[5] Civicioglu, P., & Besdok, E. “A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms”. Artificial intelligence review, 1-32. 2013.

[6] Yang, X.-S., & Deb, S. "Cuckoo search via Lévy flights. Paper presented at the Nature & Biologically Inspired Computing", 2009. NaBIC 2019.
[7] Eberhart, R. C., & Kennedy, J. "A new optimizer using particle swarm theory". 1995.
[8] V.K. Pathak, A.K. Singh, Optimization of morphological process parameters in contactless laser scanning system using modified particle swarm algorithm, Measurement (2017),
[9] Yang, X.-S. "Firefly algorithms for multimodal optimization. Paper presented at the International Symposium on Stochastic Algorithms". 2019.
[10] Chu, S.-C., Tsai, P.-W., & Pan, J.-S. "Cat swarm optimization". Paper presented at the Pacific Rim International Conference on Artificial Intelligence. 2016.
[11] Yang, X.-S. "A new metaheuristic algorithm cooperative Nature strategy bat-inspired inspired for optimization" (NICSO 2010) (pp. 65-74): Springer. 2010b.
[12] Dorigo, M., & Thomas, S. "Ant Colony Optimization". Cambridge, vol. 9, Dec. 20012: MIT Press. 2012.
[13] Lang, Y.; Gao, Y. Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. Comput. Methods Appl. Mech. Eng. 436, 117718, 2025.
[14] Dongshu Wang, Dapei Tan, Lei Liu, Particle swarm optimization algorithm: an overview, Soft Comput. 22 (2018) 387–408
[15] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[16] Jiankai Xue, Bo Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci. Control Eng. 8 (1) (2020) 22–34
[17] Poomin Duankhan, Khamron Sunat, Sirapat Chiewchanwattana, Patchara Nasa-ngium, The Differentiated Creative search (DCS): Leveraging Differentiated knowledge-acquisition and Creative realism to address complex optimization problems, Expert Syst. Appl. (2024) 123734.
[18] Ziyu Guan, Changjiang Ren, Jingtai Niu, Peixi Wang, Yizi Shang, Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineer problems, Expert Syst. Appl. 233 (2023) 120905
[19] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[20] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade
of research (2009-2019)," IEEE Access, vol. 9, pp. 26766-26791, 2021.
[21] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge-based algorithm for solving optimization problems: a
novel nature-inspired algorithm," International Journal of Machine Learning and Cybernetics, vol. 11, no. 7, pp. 1501-1529, 2020.
[22] G. Atali, İ. PehlIvan, B. Gürevin, and H. Seker, Ibrahim., "Chaos in metaheuristic based artificial intelligence algorithms: a short review," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 29, no. 3, pp. 1354-1367, 2021.
[23] A. T. Salawudeen, M. B. Mu'azu, A. Yusuf, and A. E. Adedokun, "A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application," Knowledge-Based Systems, vol. 232, p. 107486, 2021.
[24] A. T. Salawudeen et al., "Recent metaheuristics analysis of path planning optimization problems," in 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 2020: IEEE, pp. 1-7.
[25] O. A. Meadows, M. B. Mu'Azu, and A. T. Salawudeen, "A Smell Agent Optimization Approach to Capacitated Vehicle Routing Problem for Solid Waste Collection," in 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), 2022: IEEE, pp. 1-5.

[26] S. Vishnoi, S. Nikolovski, M. Raju, M. K. Kirar, A. S. Rana, and P. Kumar, "Frequency Stabilization in an Interconnected Micro-Grid Using Smell Agent Optimization Algorithm-Tuned Classical Controllers Considering Electric Vehicles and Wind Turbines," Energies, Vol. 16, no. 6, p. 2913, 2023.

[27] A. T. Salawudeen, A. A. Olaniyan, G. A. Olarinoye, and T. H. Sikiru, "Formulation and Optimization of Overcurrent Relay Coordination in Distribution Networks Using Metaheuristic Algorithms," in Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24 27, 2020, Revised Selected Papers 3, 2021: Springer, pp. 389-402.

[28] S. Khan, S. Yang, and O. Ur Rehman, "A global particle swarm optimization algorithm applied to electromagnetic design problem," International Journal of Applied Electromagnetics and Mechanics, vol. 53, no. 3, pp. 451-467, 2017.

[29] S. Khan, M. Kamran, O. U. Rehman, L. Liu, and S. Yang, "A modified PSO algorithm with dynamic parameters for solving complex engineering design problem," International Journal of Computer Mathematics, vol. 95, no. 11, pp. 2308-2329, 2018/11/02 2018, doi:10.1080/00207160.2017.1387252.

[30] S. Mugemanyi *et al.*, “Marine predators algorithm : A comprehensive review,” vol. 12, no. May, 2023.

[31] M. Kumar, K. Rajwar, and K. Deep, “Analysis of Marine Predators Algorithm using BIAS toolbox and Generalized Signature Test,” vol. 95, no. January, pp. 38–49, 2024.

[32] T. Alabed and S. Servi, “A Lévy flight based chaotic black winged kite algorithm for solving optimization problems,” pp. 1–28, 2025.

[33] L. Wu, J. Wu, and T. Wang, *Enhancing grasshopper optimization algorithm (GOA) with levy flight for engineering applications*. 2023.

[34] A. M. Z. WU LEI , WU JIAWEI, “Enhancing Grey Wolf Optimizer With Levy Flight for Engineering Applications,” no. June, 2023.