MODIFIED INDIVIDUAL EXPERIENCE MAYFLY ALGORITHM

Nicholas Kwesi **PRAH II**, Emmanuel Assuming **FRIMPONG,** Elvis **TWUMASI**
Department of Electrical and Electronic Engineering, College of Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
bravokwesi@gmail.com, eafrimpong.soe@knust.edu.gh, etwumasi.coe@knust.edu.gh.

Keywords: Optimization, Metaheuristics, Swarm Intelligence.

Abstract*: An algorithm that modifies the individual experience of mayflies in the mayfly algorithm (MA) to enhance its performance, is proposed. The proposed algorithm called the Modified Individual Experience Mayfly Algorithm (MIE-MA) calculates the experience of a mayfly by finding an average of the positions the mayfly has been to instead of just using the best position. A chaotic decreasing gravity coefficient is also employed to enhance the balance between the exploitation and exploration of the algorithm. The proposed algorithm was compared to the original MA, and two recent variants named, PGB-IMA and ModMA, on eight benchmark functions. The parameters used for comparison were Mean Absolute Error, Standard Deviation, and convergence rate. The results validate the superior performance of the MIE-MA over the other three algorithms. The MIE-MA yields better optimal values with minimal iterations.*

1. INTRODUCTION

Metaheuristic algorithms are a paradigm of computational intelligence helpful in solving complex problems. They offer advantages such as easy application to continuous and discrete problems, less complex mathematical computations, and efficient search for global optimum solutions [1]. Metaheuristic Algorithms are classified into single-solution and population-based algorithms [1, 2]. In single solution-based methods, a generated solution is improved continuously until a stopping criterion is met [2]. In population-based methods, a group of solutions is generated within a search space and is updated at each iteration to find an optimal solution [3]. The population-based techniques are divided into evolutionary algorithms and Swarm Intelligence optimization algorithms. Evolutionary Algorithms are hinged on natural genetic evolution. Examples are differential evolution and genetic algorithm [3, 4]. Swarm Intelligence optimization algorithms (SIOA) are based on the social behavior of animal groups. SIOAs are flexible and adaptive to various problems and have strong global search ability and robust performance [5]. Examples are the mayfly algorithm, particle swarm optimization, whale optimization algorithm, and crow search algorithm [6-8].

MA is one of the very recent SIOAs, inspired by the movement and mating process of mayflies [7]. It is a promising algorithm that exhibits enhanced exploration and exploitation abilities. Researchers have employed MA in solving complex problems [10-12]. In [8], MA was used to solve a 2D path planning problem of agricultural unmanned aerial vehicles (UAVs). In [9], MA was used to solve an optimal power flow problem in regulated electricity markets. The algorithm has also been used to improve the Maximum Power Point Tracking (MPPT) for photovoltaic systems [10].

Even though the MA is quite effective in solving complex problems, it has drawbacks such as premature convergence and stagnation. There is, therefore, a need for contributions to address these problems. In [7], an improved version of the MA called PGB-IMA was introduced. Here, the global best is selected from the whole mayfly population (both males and females) to enhance the exploration abilities of the MA. This improvement was effective on unimodal functions. However, it was found to converge slower or get stuck at local optima points on multimodal functions. In [11], levy flight was used to enhance the exploration abilities of the mayflies in the MA. This method, however, causes a mayfly to fly out of a search space in smaller search spaces. In [8], researchers adopted the exponent decreasing inertia weight, the adaptive Cauchy method, and an enhanced crossover operator to enhance the balance between exploitation and exploration of the MA. This variant was named ModMA. Although the ModMA improves the convergence rate, it does not eliminate trapping at the local optimum. Thus, there is still a need to improve the MA to holistically address the entrapment problem in local optimum and premature convergence to enhance its performance.

This work, therefore, aims at addressing the aforementioned deficiencies of the MA. A modified version of the MA is presented. The modification focuses on the individual experience of the mayflies and thus is called Modified Individual Experience Mayfly Algorithm (MIE-MA). The MIE-MA modifies the experience of the mayflies to enhance the movement of the mayflies to improve the convergence rate and move the mayflies out of local optima entrapment. This is done by replacing the personal best (*pbest*) in the MA with personal experience (*Pexp*). *Pexp* is the mean of all the positions a mayfly has been to. This allows all the positions a mayfly has visited to contribute equally to its experience. This allows the mayflies to exploit their search spaces well to avoid premature stagnation and skipping of the optimal solution in their search space. Consequently, the approach leads to optimal solutions with minimal iterations. Furthermore, a chaotic decreasing gravity coefficient is adopted to aid in the balance between the exploration and exploitation of the algorithm.

This paper is organized as follows: The original MA is described in Section 2, Section 3 provides the modified individual experience and the chaotic decreasing gravity coefficient. The benchmark functions and the test parameters used for testing the algorithms are presented in Section 4. Results are presented and analyzed in Section 5. Section 6 concludes the paper.

2. MAYFLY ALGORITHM

The MA takes inspiration from the way mayflies fly and mate. It combines the major advantages of PSO [3], FA [12], and GA [4]. The MA is comprised of six phases [8].

2.1. Initialization

In this phase, sets of both male and female mayflies are generated at random. The current velocity and position of the *i*th mayfly are assigned as $v_i = (v_{i1}, v_{i2} \ldots, v_{in})$ and $x_i = (x_{i1}, x_{i2} \ldots, x_{in})$, respectively. Positions of the mayflies are modified based on their best ever position (*pbest*) position and the best position in the whole population (*gbest*).

2.2. Movement of male mayflies

The positions of male mayflies are updated based on the positions of mayflies around them and its past positions. $\boldsymbol{x_i^t}$ denotes the present position of the *i*th male mayfly at iteration *t*. In order to update the position of a particular mayfly, the velocity $\boldsymbol{v_i^{t+1}}$ is added to its current position. i.e.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{1}$$

The male mayfly's velocity update is obtained from:

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-vd_p^2}\left(pbest_{ij} - x_{ij}^t\right) + a_2 e^{-vd_g^2}\left(gbest_j - x_{ij}^t\right) \tag{1.1}$$

where:

v = visibility coefficient.

a_1 and a_2 = positive attraction constants.

$pbest_{ij}$ = Mayfly ith best position in dimension j.

d_p and d_g = Euclidean distances between mayfly i and its best position and between mayfly i and the best position, respectively.

g = gravity coefficient, can be a fixed number between 0 and 1, or can be expressed as:

$$g = g_{max} - \frac{g_{max} - g_{min}}{iter_{max}} \times iter \quad (2)$$

where,

g_{max} *and* g_{min} = maximum and minimum values of g.

$iter$ = current iteration.

$iter_{max}$ = maximum no. of iterations.

The best position of the mayfly in iteration *t+1* is determined as:

$$pbest_i = \begin{cases} x_i^{t+1}, \ if\ f(x_i^{t+1}) < f(pbest_i). \\ kept\ the\ same,\ otherwise. \end{cases} \quad (3)$$

The top-performing male mayflies continue to execute oscillatory movements at varying velocities. These velocities are determined by:

$$v_{ij}^{t+1} = v_{ij}^{t} + d * r, \quad (4)$$

d = nuptial dance coefficient

r = a random value in the range [-1, 1]

2.3. Movement of female mayflies.

Female mayflies move towards male mayflies. Their positions are updated by the following formula:

$$y_i^{t+1} = y_i^{t} + v_i^{t+1} \quad (5)$$

where:

y_i^{t+1}= female mayfly i’s position in iteration t+1.

y_i^{t} = female mayfly i’s position in iteration t.

v_i^{t+1} = female mayfly i’s velocity for iteration t+1.

For minimization problems, female mayflies’ velocity updates are calculated as:

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^{t} + a_2 e^{-\beta r_d^2}\left(x_{ij}^{t} - y_{ij}^{t}\right),\ if\ f(y_i) > f(x_i). \\ g * v_{ij}^{t} + w * n,\ iff(y_i) \le f(x_i). \end{cases} \quad (6)$$

where,

v_{ij}^{t} and y_{ij}^{t} = The velocity and position of female mayfly *i* in dimension *j* during iteration *t*.

a_2 and β = constants which represent attraction and visibility, respectively.

r_d = Euclidean distance between female mayfly *i* and male mayfly *i*.

w= coefficient for random walk. Used when no attraction between male and female.

n= random number between −1 to 1.

2.4. Mating Phase

The crossover operator is used to represent this phase. Each mayfly pair produces two offspring. This is expressed as:

$$off1 = R \cdot m + (1-R) \cdot f \quad (7)$$

$$off2 = R \cdot f + (1-R) \cdot m \quad (8)$$

R = random value.
f and *m*= female and male respectively.

2.5. Mutation phase

This phase aims to enhance the exploitation abilities of the MA. This is done by mutating chosen offspring. This is expressed as:

$$offspring_n = offspring_n + \sigma N_n(0,1) \quad (9)$$

where,
σ and *Nn*= standard deviation and the standard normal distribution, respectively.

2.6. Reduction of Nuptial Dance and Random Walk

The random walk and nuptial dance are reduced through a geometric progression over the iterations in this phase. This is to aid the balance between exploration and exploitation of the MA. This is expressed as:

$$d_t = d_o \delta^t,\ 0 < \delta < 1 \quad (10)$$

$$fl_t = fl_o \delta^t,\ 0 < \delta < 1 \quad (11)$$

where, t = iteration and δ= a value between 0 and 1.

2.7. The process is summarized in the pseudocode below:

Objective function $f(x), x = (x_1, \ldots, x_d)^T$
Initialize the positions and velocities of the male mayfly and female mayfly population

Evaluate the solutions and find *gbest*
Do while iteration < maximum iterations
    Update the velocities and solutions of both male and female mayflies
    Assess the solutions
    Sort and rank the mayflies
    Mate the mayflies
    Evaluate the resulting offspring
    Randomly assign offspring to male and female categories
    Substitute inferior solutions with superior solutions
    Update the individual best (*pbest*) and global best (*gbest*) solutions
End while
Display Results

3. PROPOSED MODIFIED INDIVIDUAL EXPERIENCE MA

In the original MA, the position of each mayfly is adjusted according to its individual experience and the experience of its neighbors. The individual experience is represented as pbest which is the best position the mayfly ever visited. The deficiency with this approach is that mayflies which are averagely moving at a better rate than the gbest might not be given adequate opportunities to contribute to the global best. This approach may lead to a situation where the gbest is provided by mayflies stuck in a local optimum and hence may cause stagnation of the whole MA.

In this modification, the experience of the mayfly is represented as the mean of the positions the mayfly has been to in the search space. This provides a better representation of the experience and also provides a better picture of how the mayflies are approaching the global optimum and consequently provide the optimum value in the search space. This is formulated as:

$$P_{exp,i}^{t} = \frac{\sum_{t=1}^{iter} x_i^t}{iter} \quad (12)$$

where;

$P_{exp,i}^{t}$ = The experience of mayfly *i* at step *t*.
iter = Current iteration.
x_i^t = Position of mayfly *i* at step *t*.

Also, a chaotic random decreasing gravity coefficient strategy is adopted to enhance the balance between the exploration and exploitation abilities of the MA. This is motivated by a study in [13], where different weight strategies were applied to PSO to determine their influence. This is due to its ability to rough search and minute search alternately in all its evolutionary processes [14]. A slightly modified version is used in this work and is formulated as:

$$g = (gmax - gmin) * \left(\frac{MaxIt - iter}{MaxIt}\right) + \mathrm{gmin} * \mathrm{z} \quad (13)$$
$$z = 4 * z * (1 - z)$$

where*: gmax* & *gmin* are the maximum and minimum inertia weights respectively and z = random number between 0 and 1.

3.1. The pseudocode of the proposed modification is shown below:

Objective function $f(x), x = (x_1, \ldots, x_d)^T$
Initialize the positions and velocities of the male and female mayfly population
Evaluate solutions and find gbest
Do while iteration<maximum iterations
    Update the velocities and solutions of both male and female mayflies
    Sort and rank the mayflies
    Mate the mayflies
    Evaluate the resulting offspring
    Randomly assign offspring to male and female categories
    Substitute inferior solutions with superior solutions
    Update mayfly experience, update pbest using modified formula in equation 12
    Update gbest
    Apply chaotic decreasing gravity coefficient in equation 13
End while
Display Results

4. TESTING OF THE PROPOSED MODIFICATION

The modified MA (MIE-MA) was tested on eight benchmark functions. The benchmark functions were obtained from [7]. The test results were compared to the original MA [7], the PGB-IMA version [7], and a modified MA named ModMA[8]. The benchmark functions were picked from the various types of benchmark functions i.e unimodal (1&2), multimodal (3&4), and fixed dimensions (5 & 6- Multimodal, 7 & 8- Unimodal) and thus provide varied levels of difficulty.

The details of the functions are shown below:

Table 1. Benchmark Functions

No.	Name	Search Range	Optimal Value	Dimensions	Error Limit
1	Sphere	[-10,10]	0	30	1.0E-05
2	Zakharov	[-5,10]	0	30	1.0E-05
3	Rastrigin	[-5.12,5.12]	0	2	1.0E-05
4	Ackley	[-1,1]	0	30	1.0E-05
5	Leon	[0,10]	0	2	1.0E-05
6	Colville	[-10,10]	0	4	1.0E-05
7	Beale	[-4.5,4.5]	0	2	1.0E-05
8	Michalewicz	[0,∏]	-9.6602	10	1.0E-05

The parameters used to compare the algorithms are shown below. The same was used in [7].

Number of iterations = 2000.
Number of runs = 50.
Male Population = 20.
Female Population = 20.
gmax=0.9, *gmin*=0.2.
$a_1 = 1, a_2 = 1.5, \beta = 2, d = 0.1, fl = 0.1, g = 0.8, \delta = 0.77$

All four algorithms were run on the same computer, an Intel® Core™ i7-7500U with CPU @ 2.70 GHz 2.90GHz and 12GB RAM.

The comparative analysis parameters used were the Mean Absolute Error (MAE) and the Standard Deviation (SD). These were calculated as.

$$MAE = \frac{1}{S}\sum_{i=1}^{S} |X_{oi} - X_i| \quad (13)$$

where: S is the number of cost samples, X_{oi} is the benchmark value of the test function and X_i is the computed optimum value.

$$SD = \sqrt{\frac{\sum(X_i - \mu)^2}{S}} \quad (14)$$

μ is the mean of the total number of cost samples.

5 RESULTS AND DISCUSSIONS

5.1. Mean absolute error and standard deviation

Table 1 below shows the MAEs and the Standard Deviations of the optimal values of the MA, PGB-IMA, ModMA, and MIE-MA on the eight test functions. The MIE-MA achieved zero MAEs on five of eight test functions (Sphere, Rastrigin, Leon, Colville, and Beale), compared to ModMA's two, MA and PGB-IMA's one. Compared to the benchmark error limit of 1E-05, MIE-MA achieved lower values for 7 of eight test functions.

The MIE-MA achieved zero SDs in 7 of eight test functions compared to ModMA's two, MA and PGB-IMA's one. Hence, the MIE-MA had the overall best performance.

Table 2. MAEs and SDs of the algorithms on the test functions

Function	Algorithm	MAE	SD
Sphere	MA	4.87967E-20	1.08699E-19
	PGB-IMA	4.6517E-20	1.10929E-19
	ModMA	7.59804E-34	1.06227E-34
	MIE-MA	**0**	**0**
Zakharov	MA	0.08919371	0.636970303
	PGB-IMA	6.76556E-10	4.6853E-09
	ModMA	7.7838E-16	2.92148E-17
	MIE-MA	**4.0552E-223**	**0**
Rastrigin	MA	2.38967	1.61645
	PGB-IMA	2.45849	2.39099
	ModMA	1.19371E-12	3.80069E-14
	MIE-MA	**0**	**0**
Ackley	MA	6.06933E-11	1.20174E-10
	PGB-IMA	8.81819E-11	1.10305E-10
	ModMA	8.89955E-13	4.28014E-15
	MIE-MA	**8.70763E-16**	**0**
Leon	MA	0	0
	PGB-IMA	0	0
	ModMA	0	0
	MIE-MA	0	0
Colville	MA	2.46273E-30	4.2634E-30
	PGB-IMA	2.23259E-30	3.91469E-30
	ModMA	7.03319E-29	3.95904E-30
	MIE-MA	**0**	**0**
Beale	MA	2.989E-02	1.494E-01
	PGB-IMA	7.471E-02	2.289E-01
	ModMA	0	0
	MIE-MA	0	0
Michalewicz	MA	0.1390	1.567E-01
	PGB-IMA	0.1065	1.161E-01
	ModMA	0.5407	1.371E-01
	MIE-MA	**9.414E-02**	**1.115E-01**

5.2. Optimum Values

Table 2 compares the optimum values of the four algorithms to the benchmark optimum values on the test functions. The values in (brackets) indicate the iteration number on which zero was achieved. MIE-MA achieved benchmark optimum values in six of eight functions (Sphere, Rastrigin, Leon, Beale, Colville, and Michalewicz), compared to modMA's four, PGB-IMA and MA's two. This is a testament to the superior performance of the MIE-MA to the other three algorithms. MIE-MA also achieved better optimum values in five of eight functions (Sphere, Zakharov, Rastrigin, Ackley, and Michalewicz) compared to the other three algorithms. The table, therefore confirms the overall superior performance of the MIE-MA.

Table 3. Comparison of optimum values

No.	Function	Benchmark Value	MA	PGB-IMA	ModMA	MIE-MA
1	Sphere	0	1.93924E-23	7.01756E-25	2.62277E-37	**0**
2	Zakharov	0	2.81062E-11	6.66323E-17	2.7536E-18	**3.0885E-296**
3	Rastrigin	0	2.9849	1.9899	0(250)	**0(50)**
4	Ackley	0	3.39258E-11	3.09982E-10	2.22045E-14	**8.88178E-16**
5	Leon	0	0(98)	**0(92)**	0(255)	0(264)
6	Beale	0	0(58)	**0(55)**	0(225)	0(229)
7	Colville	0	1.91471E-29	4.43734E-30	**0(253)**	0(268)
8	Michalewicz	-9.6602	-8.9052	-9.4513	-9.4684	**-9.6602**

5.3. Convergence Rate

Figures 1 to *8* show the convergence curves of the four algorithms for the eight test functions. It can be observed that MIE-MA has better convergence rates for five of eight test functions (Sphere, Zakharov, Rastrigin & Ackley). For test functions Leon and Beale, PGB-IMA had the best convergence rate. ModMA had the best convergence rate for the Colville test function. Even though MIE-MA did not achieve the best convergence rate for Leon, Beale, and Colville, optimum values were still achieved before the 300th iteration.

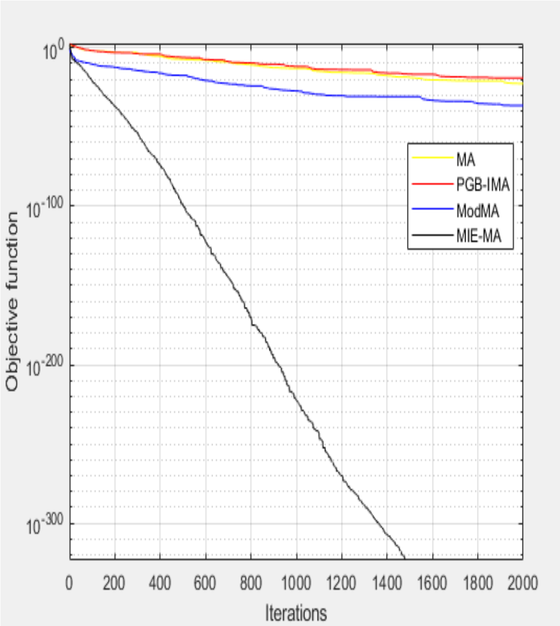

Fig. 1. Convergence Curve for Sphere

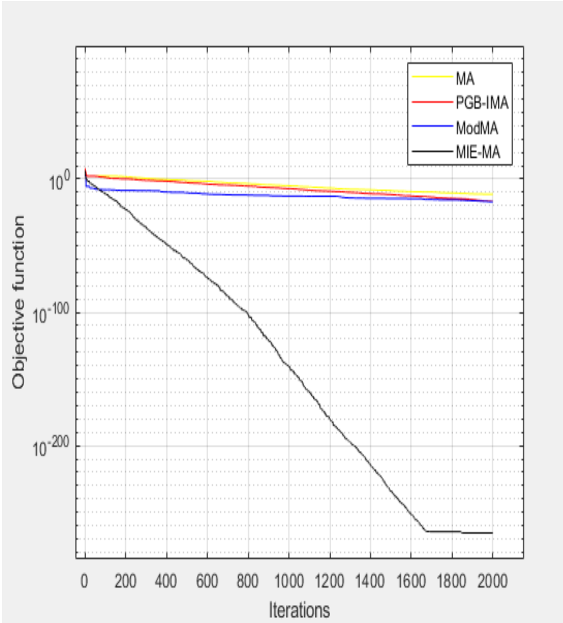

Fig. 2. Convergence Curve for Zakharov

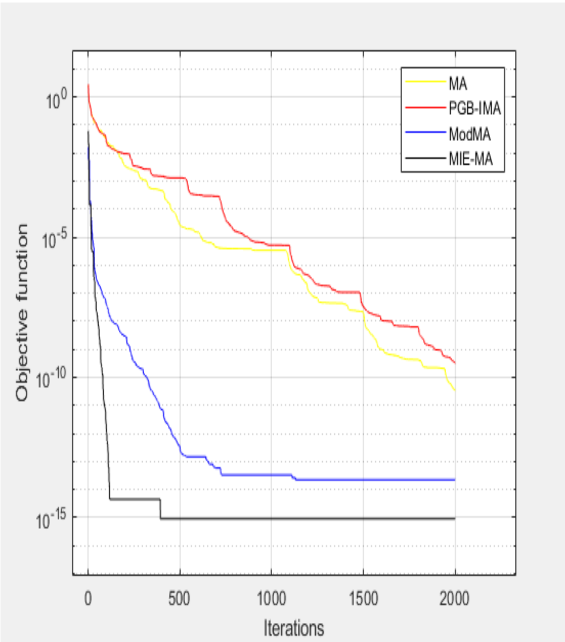

Fig. 3. Convergence Curve for Ackley

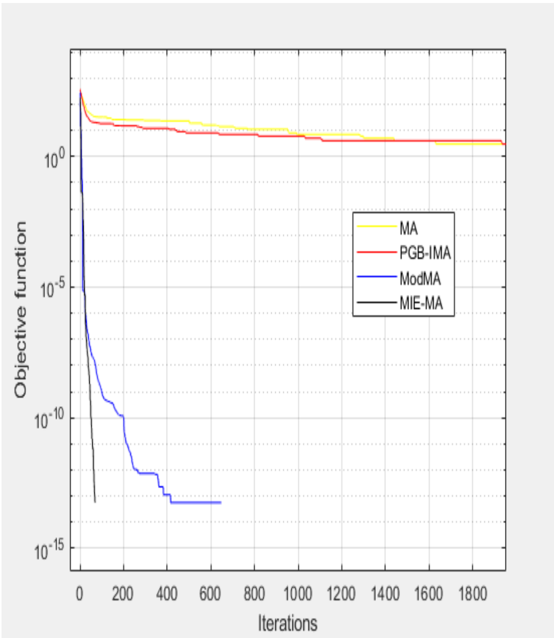

Fig. 4. Convergence Curve for Rastrigin

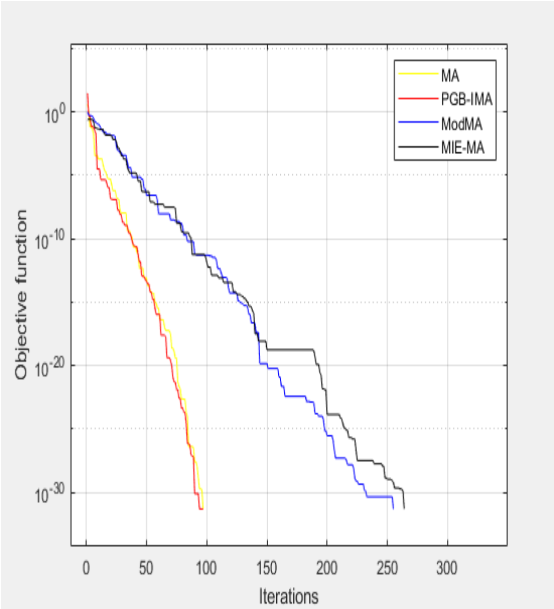

Fig.5. Convergence Curve for Leon

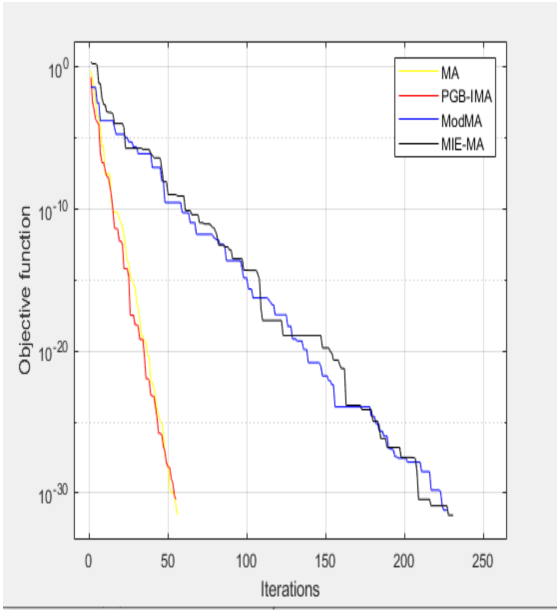

Fig. 6. Convergence Curve for Beale

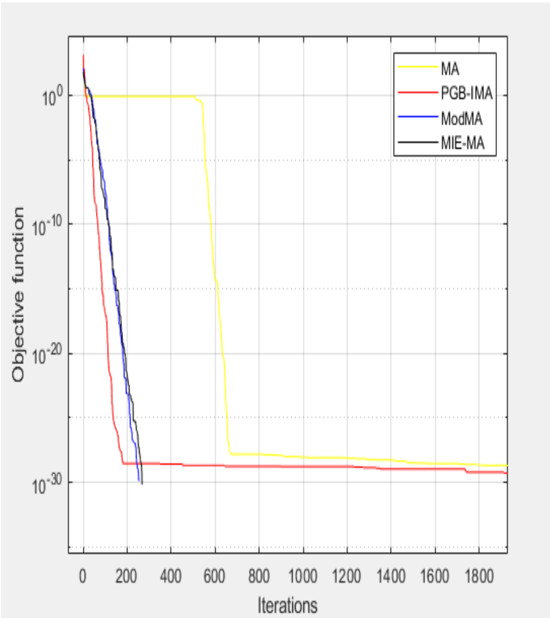

Fig. 7. Convergence Curve for Colville

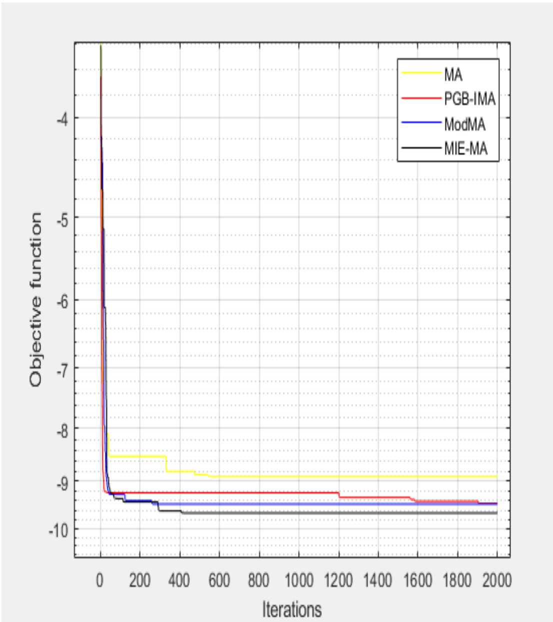

Fig. 8. Convergence Curve for Michalewicz

6. CONCLUSION

An improved mayfly algorithm called Modified Individual Experience Mayfly Algorithm (MIE-MA) has been made. This algorithm modifies the individual experience of each mayfly and also enhances the balance between exploitation and exploration. The MIE-MA achieved benchmark optimum values in six of eight test functions and also outperformed two other improvements in five out of eight functions in terms of convergence rate. The MIE-

MA also had the best MAE and SD in all eight test functions compared to the other two improvements. The results also indicated the ability of the MIE-MA to avoid local Stagnation.

REFERENCES

[1] Z. Beheshti and N. Branch, *A Review of Population-based Meta-Heuristic Algorithm*, International Journal of Advances in Soft Computing and its Applications, 5(1), pp. 1-35, 2013.

[2] J. Andres, M. Perez, N. Mladenovic, B. M. Batista, and I. J. Garcia, *Chapter 4 Variable Neighbourhood Search*, January, pp. 0–17, 2006, doi: 10.1007/0-387-33416-5.

[3] C. Wang and W. Song, *A modified particle swarm optimization algorithm based on velocity updating mechanism*, *Ain Shams Eng. J.*, vol. 10, no. 4, pp. 847–866, 2019, doi: 10.1016/j.asej.2019.02.006.

[4] D. J. Murray-Smith, *Experimental modelling: system identification, parameter estimation and model optimisation techniques*, *Model. Simul. Integr. Syst. Eng.*, pp. 165–214, 2012, doi: 10.1533/9780857096050.165.

[5] A. Banerjee, I. Nath, and B. Data, *Impacts of metaheuristic and swarm in- telligence approach in optimization Bio-Inspired Computation for Solving the Optimal Coverage Problem in Wireless Sensor Networks*, 2022.

[6] A. Askarzadeh, *A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm*, *Comput. Struct.*, vol. 169, pp. 1–12, 2016, doi: 10.1016/j.compstruc.2016.03.001.

[7] K. Zervoudakis and S. Tsafarakis, *A mayfly optimization algorithm*, *Comput. Ind. Eng.*, vol. 145, no. September 2019, p. 106559, 2020, doi: 10.1016/j.cie.2020.106559.

[8] X. Wang, J. Pan, Q. Yang, L. Kong, V. Snášel, and S. Chu, *Modified Mayfly Algorithm for UAV Path Planning*, pp. 1–21, 2022.

[9] V. B. K, S. Ramesh, P. Chandrasekar, K. Karunanithi, and A. Raja, *An improved mayfly algorithm based optimal power flow solution for regulated electric power network*, vol. 9, no. 92, pp. 979–995, 2022.

[10] S. Mo, Q. Ye, K. Jiang, X. Mo, and G. Shen, *An improved MPPT method for photovoltaic systems based on mayfly optimization algorithm*, *Energy Reports*, vol. 8, pp. 141–150, 2022, doi: 10.1016/j.egyr.2022.02.160.

[11] K. Nagarajan, A. Rajagopalan, S. Angalaeswari, L. Natrayan, and W. D. Mammo, *Combined Economic Emission Dispatch of Microgrid with the Incorporation of Renewable Energy Sources Using Improved Mayfly Optimization Algorithm*, vol. 2022, 2022.

[12] H. S. Chuah, L. P. Wong, and F. H. Hassan, *Swap-based discrete firefly algorithm for traveling salesman problem*, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10607 LNAI, no. 2019, pp. 409–425, 2017.

[13] A. Rathore and H. Sharma, *Review on inertia weight strategies for particle swarm optimization*, *Adv. Intell. Syst. Comput.*, vol. 547, no. April, pp. 76–86, 2017.

[14] Y. Feng, G. F. Teng, A. X. Wang, and Y. M. Yao, *Chaotic inertia weight in particle swarm optimization*, *Second Int. Conf. Innov. Comput. Inf. Control. ICICIC 2007*, pp. 7–10, 2007.