ISSN 2007-9737

A Hybrid Enhanced Mayfly Optimization Algorithm with Improved Performance through Fuzzy-Based Automatic Parameter Adaptation

Enrique Lizarraga, Fevrier Valdez, Patricia Melin, Oscar Castillo*

Instituto Tecnológico de Tijuana,
Mexico

ocastillo@tectijuana.mx

Abstract. Inspired by the unique behavioral patterns of mayflies, characterized by their brief lifespans and complex mating dynamics, the Mayfly algorithm represents a novel and effective optimization approach. Rooted in the principles of particle swarm optimization, this algorithm combines swarm intelligence with evolutionary mechanisms to achieve enhanced performance in solving computational problems. This study focuses on improving the Mayfly algorithm through the adaptive adjustment of its parameters, leveraging fuzzy logic for stability in exploration and exploitation. The proposed adaptation enhances the algorithm's capability to address optimization tasks, demonstrating superior performance in convergence speed and solution reliability. Simulation results show the advantages of the hybrid approach.

Keywords. Mayfly algorithm, evolutionary algorithms, fuzzy parameter adaptation, optimization techniques, exploration and exploitation, genetic algorithms.

1 Introduction

In everyday life, the decisions we make, consciously or unconsciously, often aim at maximizing benefits or minimizing losses. This process of selecting the best option among various alternatives is known as optimization [1]. Its goal is to find the most effective possible outcome, whether by maximizing or minimizing a result, with or without defined constraints. Optimization includes key elements, involving decision elements, constraints, and purposes, and it is essential for solving problems in various fields [2].

However, traditional nature inspired optimization methods, such as particle swarm optimization (PSO) or other swarm intelligence algorithms, often face challenges in handling high dimensional spaces or avoiding local optima, thus producing subpar performance in complicated situations [3].

This work aims at tackling these limitations by introducing a hybrid approach that enhances algorithm performance in such environments.

In optimization, the primary goal is to search within a solution space to identify the most advantageous option from a range of possibilities [4].

Often, the search space is vast, resulting in extended time requirements for finding an optimal solution [5]. To tackle this issue, computational intelligence methods have been developed to enhance the process of finding a solution for optimization and search problems [6]. These approaches provide competitive outcomes, although not always the best ones. Alternatively, metaheuristic algorithms serve as effective methods, as they can identify near optimal solutions in a realistic timeframe, even without guaranteeing the absolute best result [7].

Metaheuristics are grouped into three main types: evolutionary, based on physics, chemistry, and swarm intelligence [8]. Evolutionary methods, like genetic or differential evolution algorithms, are inspired by natural selection. Physics- and chemistry-based algorithms mimic natural laws, such as the Big Bang Crunch algorithm. Swarm intelligence, or collective intelligence, draws from natural group behaviors like ant colonies or bird flocks and is widely researched to approach optimization challenges [9].

In this context, the Mayfly Algorithm (MA) is a modification of PSO (Kennedy & Eberhart, 1995), combining the strengths of PSO, genetic algorithms (GA), and swarm intelligence-based algorithms [10].

Inspired by mayfly behavior, the Mayfly Algorithm uses techniques, like genetic crossover and local search to enhance PSO's performance, in complex multidimensional scenarios where adjustments are required to ensure optimal solutions [11]. This hybrid approach follows the trend of modified algorithms that combine the advantages of existing methods, which have been widely studied in the literature [12].

Fuzzy sets were first introduced by Zadeh in 1965, with Mamdani being the pioneer in applying them to control in 1974. Both advancements enabled fuzzy controllers to be effectively implemented in various applications [13]. Fuzzy logic offers a systematic computation framework to handle linguistic information and enhances numerical computation by using linguistic labels defined through membership functions (MFs) [14].

This study introduces a fuzzy adaptation approach to the position update MA parameters, aiming to improve its effectiveness in optimizing various mathematical functions. By utilizing different fuzzy rules, it is possible to address stagnation in local optima, which is a common issue in swarm intelligence algorithms, such as PSO, ACO, and others. This strategy finds an effective balance between exploration and exploitation, leading to improved results.

Mayfly with fuzzy parameter adaptation better addresses high dimensionality problems due to its ability to adapt to noise with the help of our fuzzy rules, unlike other classic methods such as PSO.

The article is structured as: Section 2 offers the theoretical framework, including background of the Mayfly algorithm. Section 3 outlines the proposed method through a flowchart. In Section 4, the design of the fuzzy adapter is described in detail. Section 5 summarizes the results, and Section 6 evaluates the parameters that have a significant influence on MA. Section 7 offers discussion of results and Section 8 the conclusions.

2 Materials and Methods

Evolutionary techniques, including techniques like genetic algorithms (GAs) and PSO, are powerful tools to address complex problems. PSO, inspired by bird flock behavior, balances exploration and exploitation in its seek optimal solutions, gaining popularity for its effectiveness in applications related to science and engineering [15].

On the other hand, evolutionary algorithms emulate biological processes, like mutation, recombination, and natural selection to improve solutions in a search space [16].

Genetic algorithms, introduced by John Holland, work with populations of solutions encoded as chromosomes, using genetic operators to continuously refine answers. These techniques stand out for their capacity to identify the best solutions in complex multimodal spaces efficiently and adaptively [17].

The MA is a method based on the social behavior of mayflies (Ephemeroptera). These insects spend most of their life as aquatic nymphs, but once they mature, males gather in swarms to perform a "nuptial dance" in the air to attract females, who then lay eggs in the water. This mating process serves as the inspiration for the algorithm, which simulates the search for ideal solutions in multi-dimensional environments [18].

2.1 PSO Algorithm

The main idea of the algorithm is to simulate social behaviour as particles in a solution space, where each particle represents a potential solution. The position of each particle is adjusted based on its personal better position (pbest) and the best global position (gbest) of the swarm [9].

In this approach, the control of velocity is regarded as a key element, as it serves as the primary mechanism for adjusting a particle's position to explore the solution domain in pursuit of an optimal solution [19].

Eberhart analyzed the maximum velocity values and evaluated outcomes across various velocity settings. The velocity of particle k within the swarm is updated during the (i+1)th step calculated using the equation below:

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_1^t y_{ij-}^t x_{ij}^t \ + c_2 r_2^t y_{ij-}^t x_{ij}^t, \quad (1)$$

where vij(t) is the velocity of particle i in dimension j at time t, c1 is the cognitive factor (importance of the particle's best previous position), c2 is the social factor (importance of the swarm's best global position), r1 and r2 represent randomly generated numbers within the specified limits [0, 1], yj(t) Is the best position of particle i in dimension j,xij(t) Is the new position of particle i in dimension j,yj(t) is the best global position of the swarm in dimension j.

The position of each particle i is updated at every (i+1)-th iteration based on the equation:

$$x_k{}^{t+1} = x_k{}^{t} + v_k{}^{t+1}, \qquad (2)$$

where xi is the position of particle k, t is the current iteration, vi is the velocity of particle k [20].

2.2 Mayfly Algorithm

Mayfly is an insect with a clear purpose, to reproduce before extinguishing; therefore, the Mayfly algorithm is inspired by the frantic search for a mate by Mayfly insects, enhancing the convergence of PSO by constantly updating particle positions, thus achieving a more efficient and faster search for optimal solutions [21]:

$$\text{xi t+1} = \text{xti} + \text{vi t+11}, \qquad (3)$$

$$\text{vij t+1} = \text{vij t} + \text{dr}, \qquad (4)$$

$$\begin{aligned} v_{ij}^{t+1} = v_{ij}^{t} + a_1 e^{-\beta r_p^2}\left(\text{pbest}_{ij} - x_{ij}^{t}\right) \\ + a_2 e^{-\beta r_g^2}\left(\text{gbest}_j - x_{ij}^{t}\right), \end{aligned} \qquad (5)$$

where Vtij is the velocity of Mayfly i in dimension j=1-n at time step t,$x\ t\ ij$ is the position of ephemeral i in dimension j at time step t, a1 y a2 are positive attraction constants used to scale the contribution of the cognitive and social components, respectively, pbest: is the best local position, gbest is the best global position , β is a fixed visibility coefficient [22].

In contrast to males, female mayflies do not form swarms. Instead, they move toward males for mating purposes. Assuming that y_i^t it represents the recent position of a female mayfly in the problem domain at time step t, her position is refreshed by adding a velocity component to the current position, as follows:

$$y_i^{t+1} = y_i^t + v_i^{t+1}, \qquad (6)$$

$$v_{ij}^{t+1} = \begin{matrix} v_{ij}^t + a_2 e^{-\beta r_{mf}^2}\left(x_{ij}^t - y_{ij}^t\right), & if\ (f(y_i) > f(x_i)) \\ v_{ij}^t + flr, & if\ (f(y_i) \le f(x_i)) \end{matrix}, \qquad (7)$$

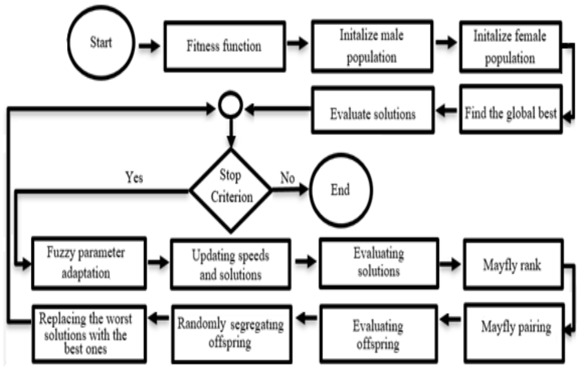
Start
Fitness function
Initalize male population
Initalize female population
Find the global best
Evaluate solutions
Stop Criterion
Yes
No
End
Fuzzy parameter adaptation
Updating speeds and solutions
Evaluating solutions
Mayfly rank
Mayfly pairing
Evaluating offspring
Randomly segregating offspring
Replacing the worst solutions with the best ones

Fig. 1. Fuzzy parameter adaptation in MA

where v_ij^t denotes the velocity of the female mayfly in dimension j at time step t, and y_ij^t denotes her position in the same dimension and time step. The parameter a2 is a positive attraction constant, and β is a fixed visibility coefficient [18].

The distance between the male and female mayflies, rmf. Lastly, fl is a random walk coefficient used when the female is not drawn toward a male, causing her to fly randomly, with r being a random value within a specified range (-1, 1) [23].

2.3 Dynamic Elite Strategy Mayfly Algorithm (DESMA)

The DESMA algorithm employs an elite selection strategy around the global optimal solution to avoid local optima, enrich population diversity, and expand the search scope [24].

The algorithm adjusts the search range derived from the evaluation between the current global optimal solution and previous generations. If the current solution is superior to the previous one, the search area expands; otherwise, it contracts to maintain the previous global optimum [24]:

$$\text{R}=\begin{cases} \text{R} * \text{c}_1, \text{if f(cgbest)} < \text{f(lgbest)} \\ \text{R} * \text{c}_2, \text{if f(cgbest)} \ge \text{f(lgbest)} \end{cases}, \qquad (8)$$

where R is the search range in a particular space, c1 is an increase factor, set to 1.05, c2 is a reduction factor, set to 0.95, cgbest is the current global optimal position, and lgbest is the previous optimal position:

$$\text{r1} = 2 * \text{rand}(1, \text{n}) - 1, \qquad (9)$$

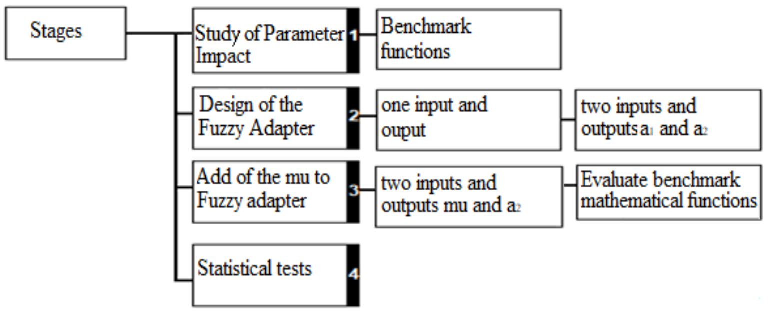
Stages
Study of Parameter Impact
1
Benchmark functions
Design of the Fuzzy Adapter
2
one input and ouput
two inputs and outputs a₁ and a₂
Add of the mu to Fuzzy adapter
3
two inputs and outputs mu and a₂
Evaluate benchmark mathematical functions
Statistical tests
4

Fig. 2. Stages of the Fuzzy Adaptation Development

Table 1. Test functions

Function	Name
F1	Sphere
F2	Rastrigin
F3	Ackley
F4	Griewank
F5	Schwefel
F6	Powell
F7	Rosenbrock
F8	Alpine
F9	Zakharov
F10	Sum squares
F11	Styblinskitang

where r_1 is a random number in [−1,1], and n is the dimension:

$$\text{egbest}=\text{egbest} + \text{r1} * \text{R}, \tag{10}$$

where egbest is the elite mayfly within the search range, and cgbest is the current global optimum [24].

2.4 Modified Mayfly Algorithm (MODMA)

The original MA algorithm shows fast convergence, but struggles with local optima [19]. To address this, the MODMA algorithm enhances the exploration capability and convergence speed of MA through a new crossover operator, a strategy to balance exploration and exploitation, and adaptive Cauchy mutation to increase diversity:

$$\text{g} = \text{g}_{\text{min}} + \exp\left(1 - \frac{\text{iter}_{\text{max}}}{\text{iter}_{\text{max}} - \text{iter} + 1}\right), * (\text{g}_{\text{max}} - \text{g}_{\text{min}}), \tag{11}$$

where gmax and gmin denote the largest and smallest weights, In sequence. The present and overall number of cycles are expressed as iter and itermax, sequentially.

This algorithm improves by employing horizontal crossover in order to enhance exploration capability and improve convergence speed [26].

$$\begin{cases} r1 = L * \ male + (1 - L) * female + \\ \quad c_1 * (male - female), \\ r2 = L * \ female + (1 - L) * male + \\ \quad c_2 * (female - male), \end{cases} \tag{12}$$

where c1 and c2 are arbitrary values between -1 and 1 generated by a consistent distribution. The horizontal crossover operation divides the multidimensional problem-solving space into

Table 2. Analysis of the a_1 parameter for 50 dimensions

Function		a_1=1	a_1=3	a_1=10
F1	x̄	$2.662\text{x}10^{-18}$	$8.606\text{x}10^{-18}$	$3.152\text{x}10^{-18}$
	s	$4.304\text{x}10^{-18}$	$2.117\text{x}10^{-17}$	$7.270\text{x}10^{-18}$
F2	x̄	$2.036\text{x}10^{1}$	$2.023\text{x}10^{1}$	$2.001\text{x}10^{1}$
	s	$7.531\text{x}10^{0}$	$5.771\text{x}10^{0}$	$3.972\text{x}10^{0}$
F3	x̄	$3.111\text{x}10^{0}$	$3.123\text{x}10^{0}$	$3.125\text{x}10^{0}$
	s	$4.198\text{x}10^{-2}$	$3.947\text{x}10^{-2}$	$3.641\text{x}10^{-2}$
F4	x̄	$1.070\text{x}10^{-17}$	$2.624\text{x}10^{-9}$	$5.344\text{x}10^{-18}$
	s	$2.294\text{x}10^{-17}$	$1.437\text{x}10^{-8}$	$1.020\text{x}10^{-17}$
F5	x̄	$9.866\text{x}10^{-4}$	$5.766\text{x}10^{-4}$	$1.600\text{x}10^{-3}$
	S	$3.212\text{x}10^{-3}$	$5.766\text{x}10^{-4}$	$4.100\text{x}10^{-3}$
F6	x̄	$2.075\text{x}10^{4}$	$2.075\text{x}10^{4}$	$2.075\text{x}10^{4}$
	S	$2.686\text{x}10^{0}$	$3.007\text{x}10^{0}$	$2.730\text{x}10^{0}$
F7	x̄	$1.562\text{x}10^{3}$	$1.592\text{x}10^{3}$	$1.626\text{x}10^{3}$
	s	$6.374\text{x}10^{2}$	$6.418\text{x}10^{1}$	$5.569\text{x}10^{1}$
F8	x̄	$1.242\text{x}10^{-31}$	$1.005\text{x}10^{-31}$	$7.025\text{x}10^{-32}$
	s	$2.916\text{x}10^{-31}$	$1.450\text{x}10^{-31}$	$1.168\text{x}10^{-31}$
F9	x̄	$8.458\text{x}10^{1}$	$9.203\text{x}10^{1}$	$5.183\text{x}10^{1}$
	s	$3.895\text{x}10^{1}$	$4.820\text{x}10^{1}$	$1.735\text{x}10^{1}$
F10	x̄	$4.682\text{x}10^{-78}$	$3.052\text{x}10^{-97}$	$1.903\text{x}10^{-86}$
	s	$8.549\text{x}10^{-79}$	$1.658\text{x}10^{-96}$	$1.042\text{x}10^{-85}$

Table 3. Analysis of the a_2 parameter for 50 dimensions

Function		a_2=1	a_2=3	a2=10
F1	x̄	$7.770\text{x}10^{-18}$	$6.007\text{x}10^{-1}$	$1.143\text{x}10^{1}$
	s	$1.491\text{x}10^{-17}$	$4.835\text{x}10^{-1}$	$3.718\text{x}10^{0}$
F2	x̄	$2.148\text{x}10^{1}$	$1.041\text{x}10^{2}$	$3.291\text{x}10^{2}$
	s	$1.133\text{x}10^{1}$	$6.560\text{x}10^{1}$	$5.415\text{x}10^{1}$
F3	x̄	$2.148\text{x}10^{1}$	$3.421\text{x}10^{0}$	$4.267\text{x}10^{0}$
	s	$1.135\text{x}10^{1}$	$3.042\text{x}10^{1}$	$3.324\text{x}10^{-1}$
F4	x̄	$4.991\text{x}10^{-18}$	$3.205\text{x}10^{-1}$	$1.055\text{x}10^{1}$
	s	$1.384\text{x}10^{-17}$	$3.501\text{x}10^{-1}$	$3.465\text{x}10^{0}$
F5	x̄	$9.031\text{x}10^{-4}$	$1.000\text{x}10^{-2}$	$3.2400\text{x}10^{-1}$
	s	$3.500\text{x}10^{-3}$	$1.150\text{x}10^{-2}$	$8.260\text{x}10^{-2}$
F6	x̄	$2.075\text{x}10^{4}$	$2.075\text{x}10^{4}$	$2.075\text{x}10^{4}$
	s	$3.693\text{x}10^{0}$	$3.730\text{x}10^{0}$	$3.133\text{x}10^{0}$
F7	x̄	$1.751\text{x}10^{3}$	$1.689\text{x}10^{3}$	$1.488\text{x}10^{3}$
	s	$5.053\text{x}10^{1}$	$6.103\text{x}10^{1}$	$7.758\text{x}10^{1}$
F8	x̄	$4.910\text{x}10^{-30}$	$4.060\text{x}10^{-2}$	$1.117\text{x}10^{-1}$
	s	$1.134\text{x}10^{-29}$	$2.238\text{x}10^{-2}$	$1.065\text{x}10^{-1}$
F9	x̄	$3.450\text{x}10^{-15}$	$7.049\text{x}10^{0}$	$2.486\text{x}10^{2}$
	s	$3.106\text{x}10^{-15}$	$8.0035\text{x}10^{0}$	$8.5808\text{x}10^{1}$
F10	x̄	$1.852\text{x}10^{-155}$	$1.183\text{x}10^{-8}$	$2.486\text{x}10^{2}$
	s	$1.041\text{x}10^{-154}$	$2.700\text{x}10^{-8}$	$8.580\text{x}10^{1}$

median populations of hypercubes, allowing for sampling of new points on the periphery of the hypercube with low probability [25].

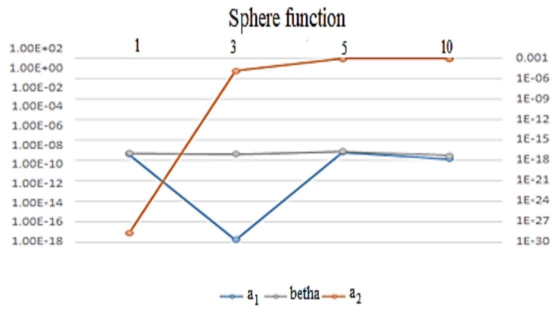

Fig. 3. Impact of parameters on the sphere function

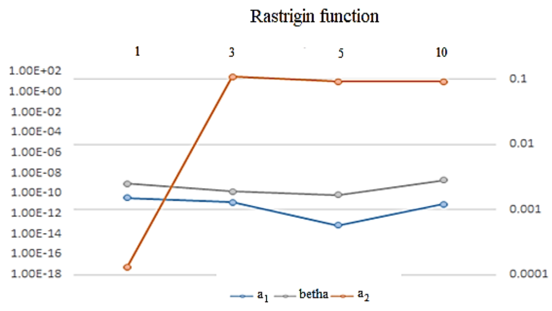

Fig. 4. Impact of parameters on the Rastrigin function

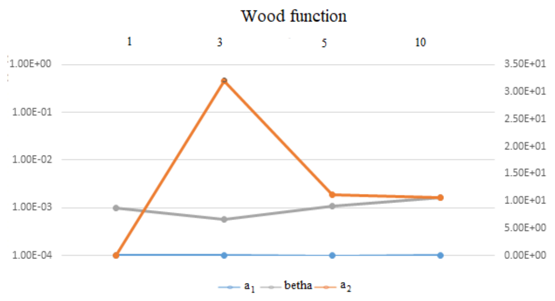

Fig. 5. Impact of parameters on the wood benchmark function

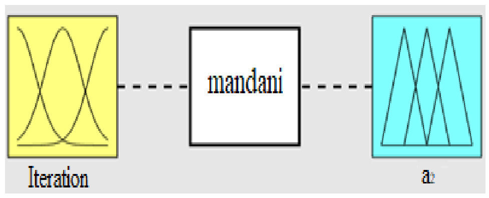

Fig. 6. Fuzzy adapter structure

Table 4. Analysis of the β parameter for 50 dimensions

Function		β =1	β =3	β =10
F1	x̄	2.662×10^{-18}	8.606×10^{-18}	3.152×10^{-18}
	s	4.304×10^{-18}	2.117×10^{-17}	7.270×10^{-18}
F2	x̄	2.036×10^{1}	2.023×10^{1}	2.001×10^{1}
	s	7.531×10^{0}	5.771×10^{0}	3.972×10^{0}
F3	x̄	3.111×10^{0}	3.123×10^{0}	3.125×10^{0}
	s	4.198×10^{-2}	3.947×10^{-2}	3.641×10^{-2}
F4	x̄	1.070×10^{-17}	2.624×10^{-9}	5.344×10^{-18}
	s	2.294×10^{-17}	1.437×10^{-8}	1.020×10^{-17}
F5	x̄	9.866×10^{-4}	5.766×10^{-4}	1.600×10^{-3}
	s	3.212×10^{-3}	2.219×10^{-3}	4.100×10^{-3}
F6	x̄	2.075×10^{4}	2.075×10^{4}	2.075×10^{4}
	s	2.686×10^{0}	3.007×10^{0}	2.730×10^{0}
F7	x̄	1.562×10^{3}	1.592×10^{3}	1.626×10^{3}
	s	6.374×10^{2}	6.418×10^{1}	5.569×10^{1}
F8	x̄	1.242×10^{-31}	1.005×10^{-31}	7.025×10^{-32}
	s	2.916×10^{-31}	1.450×10^{-31}	1.168×10^{-31}
F9	x̄	8.458×10^{1}	9.203×10^{1}	5.183×10^{1}
	s	3.895×10^{1}	4.820×10^{1}	1.735×10^{1}
F10	x̄	4.682×10^{-78}	3.052×10^{-97}	1.903×10^{-86}
	s	8.549×10^{-79}	1.658×10^{-96}	1.042×10^{-85}

2.5 Mayfly Modification (MMFO) Algorithm

The MMFO algorithm introduces a novel metaheuristic approach to problem solving. In response to observed limitations in the convergence of the conventional Mayfly algorithm mechanism is proposed that adjusts candidate positions if they overlap with both the best and global positions [27].

Candidates with velocities approaching zero cease flying and converge to the best position, although not always the global optimum, which may lead to early convergence.

Additionally, this algorithm introduces a novel strategy for updating candidate positions, inspired by the whale hunting process as per the whale optimization algorithm, where whales pursue their prey by spiraling around it. By integrating this mechanism into the MFO algorithm, the following is obtained [28]:

$$x_{ij} = (t+1) = De^{bl}\cos(2\pi l) + x_{ij}(t), \tag{13}$$

$$\mathrm{D} = \left|\mathrm{x_{ij}(t) - x_{kj}(t)}\right|, \tag{14}$$

where "b" is a constant influencing the spiral's structure, "l" represents a random value between 0 and 1, and "D" indicates the distance of the i-th individual to the best solution, establishes a spiral strategy for the prospective solution adjacent to the best, keeping it centered [27].

2.6 OCMA Algorithm

To enhance diversity and efficiency, OCMA incorporates mutation strategies. It utilizes Opposition-based Learning (OBL) and Cauchy mutation to mutate the global optimum solution, while artificial mutation operates on the offspring [29].

Inspired by OBL, a hybrid mutation strategy is cascaded to propose opposition-based solutions, preventing local optima, Cauchy mutation reduces the chance of falling into local optima. The integration of OBL and Cauchy mutation in the updated positions of mayflies forms the OCMA algorithm, aiming for improved exploration and convergence [30]:

$$x_{gbest}^{t'} = ub + r_3 *(lb - x_{gbest}^{t}), \tag{15}$$

Table 5. Fuzzy rules

Number rule	Rule description
1	IF iteration is very small then a_2 is very big
2	IF iteration is small then a_2 is big
3	IF iteration is medium then a_2 is medium
4	IF iteration is big then a_2 is small
5	IF iteration is very big then a_2 is very small

Table 6. Membership functions

Mfs	a	b	c
IMP(Very tiny iteration)	0.00	0.00	0.21
IP(Tiny iteration)	0	0.22	0.50
IM(Médium iteration)	0.28	0.54	0.78
IG(Big iteration)	0.53	0.75	1
IMG(Very big iteration)	0.75	1	1

$$x_{new}^{t+1} = a_4 *\left(x_{gbest}^{t} - x_{gbest}^{t'}\right), \tag{16}$$

where $\mathrm{x_{gbest}^{t}}$ represents the global best result of each step, x_gbest^(t^') is the solution based on the opposition of x_gbest^t, ub and lb are the upper and lower bounds of the parameters, r3 is a uniformly distributed random matrix, a4 is the coefficient of information exchange, and x_new^(t+1)is the new target solution [29].

3 Proposal

In this Section, we find an illustration of the proposal to enhance the convergence of the MA (Fig. 1).

In Fig. 1 we illustrate the Mayfly algorithm, where we first initialize the population, search for global solutions, evaluate according to the termination criterion, update velocities and solutions, then reproduce, reevaluate, and update the solution. In this new method, fuzzy adaptation of a_2 and mu is introduced to escape local minima.

Evaluating the impact of the main parameters, a_1, a_2, and β is the first step in this research. This involves analyzing various dimensions and benchmark functions, followed by plotting the results to identify the parameters that significantly influence the algorithm's performance.

Table 7. Fuzzy rules for two-input, two-output fuzzy adapter

Number rules	Fuzzy rules description
1	IF iteration is small and diversity is low then a_1 is low y a_2 is high.
2	IF iteration is small and diversity is medium then a_1 is medium low and a_2 is medium high.
3	IF iteration is small and diversity is high then a_1 is medium and a_2 is medium
4	IF iteration is medium and diversity is low then a_1 es media baja and a_2 is medium high
5	IF iteration is medium and diversity is medium then a_1 is medium and a_2 is medium.
6	IF iteration is medium and diversity is high then a_1 is medium high and a_2 is medium low
7	IF iteration is high and diversity is low then a_1 is medium and a_2 is medium.
8	IF iteration is high and diversity is medium then a_1 is medium high and a_2 is medium low.
9	IF iteration is high and diversity is high then a_1 is high and a_2 is low.

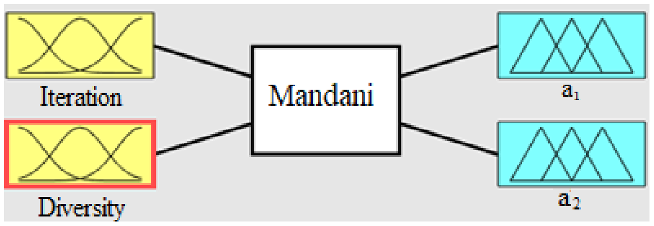

Fig. 7. Fuzzy adapter with two inputs and two outputs

Table 8. Input membership functions for the two-input fuzzy adapter

Mfs	a	b	c	d
IB (Low iteration)	0	0	0.1	0.33
IM (Medium iteration)	0.24	0.41	0.57	0.76
IA (High iteration)	0.66	0.91	1	1
DB (Low diversity)	0	0	0.1	0.33
DM (Medium diversity)	0.24	0.43	0.59	0.74
DA (High diversity)	0.66	0.89	1	1

Table 9. Output MFs for the two-input fuzzy adapter

Mfs	a	B	c	d
A1L(a_1 low)	1.09	1.24	1.27	1.42
A1ML(a_1 medium low)	1.27	1.39	1.48	1.61
A1M(a_1 medium)	1.47	1.57	1.65	1.77
A1MH(a_1 medium high)	1.64	1.76	1.84	1.98
A1H(a_1 high)	1.83	1.98	2.01	2.16
A2L(a_2 low)	0.42	0.58	0.62	0.78
A2ML(a_2 medium low)	0.62	0.78	0.82	0.97
A2M(a_2 medium)	0.85	0.97	1.02	1.15
A2MH(a_2 medium high)	1.02	1.18	1.22	1.38
A2B(a_2 Big)	1.22	1.38	1.42	1.58

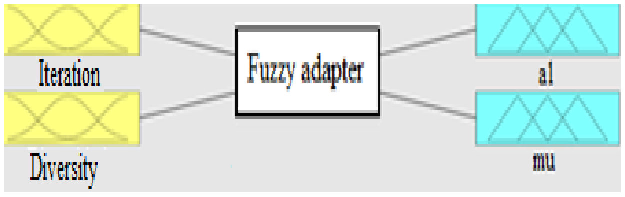

Fig. 8. Fuzzy adapter with two inputs and two outputs (a_1 and mu)

Table 10. Input MFs for the mu fuzzy adapter

Mfs	a	b	c	d
IB (small iteration)	0	0.1	0.23	0.32
IM(medium iteration)	0.25	0.42	0.56	0.67
IA (big iteration)	0.63	0.74	0.86	1
DB(small diversity)	0	0.13	0.24	0.34
DM(medium diversity)	0.27	0.42	0.54	0.64
DA (big diversity)	0.58	0.71	0.83	1

Table 11. Output MFs for the mu fuzzy adapter

MFs	A	b	c	d
A1S (a1 small)	1.20	1.33	1.36	1.49
A1MS(a1moderately small)	1.36	1.47	1.55	1.66
A1M (a1 moderate)	1.53	1.62	1.70	1.80
A1MA(a1 moderately big)	1.69	1.79	1.86	1.98
A1B (a1 big)	1.85	1.98	2.01	2.14
MuB (a2 small)	0	0.00	0.01	0.02
MuMB(mu moderately small)	0.01	0.02	0.03	0.03
MuM (mu moderate)	0.03	0.04	0.04	0.05
MuMA(mu moderately big)	0.05	0.05	0.06	0.07
MuA (mu big)	0.06	0.07	0.08	0.09

This process helps in selecting one or more of these parameters for designing the fuzzy rules [31]. When designing the fuzzy parameter adapter, it is essential to go through several testing phases. Starting with a complex adapter is not feasible, as it could lead to unnecessary or overly complicated calculations without yielding improved results.

Therefore, it is recommended to begin with a simple adapter featuring one input and one output, as well as triangular MFs [32].

One challenge we face when implementing fuzzy adaptation is that membership functions cannot be fully optimized through trial and error due to the excessive time required to test all possibilities, which is due to the complexity of variations and the algorithm's execution time.

Therefore, the use of evolutionary algorithms for this purpose is often a convenient approach [33].

In Fig. 2, we can find the outline of the process carried out for the design of our fuzzy adapters, where it is shown that the process begins with a study of the parameter impact through evaluations with benchmark functions.

For this, the dimensions and values of each of the parameters involved in the position update are varied.

4 Impact of the Parameters

This section evaluates the influence on MA performance, when varying its main velocity update components, namely the a1, a2, and β parameters.

For this study the parameter values were set as follows: β =1,a1=1.5 and a2=1.5 so that they remained in the same configuration as the original Mayfly [18].

The solution range for out test with mathematical functions was set to [-10,10] for 50 dimensions, using a 5th generation i7 computer with 8gb of RAM, the implementation of mayfly and the experiments were conducted in MATLAB, with a total of 30 executions from which the average and standard deviation were calculated.

Table 1 presents the main mathematical functions that will be used, along with the corresponding names assigned to each for the sake of simplifying the subsequent tables.

Table 2 shows the analysis of the a1 parameter for 50 dimensions.

Table 3 lists the results for the a2 parameter for 50 dimensions. Table 4 summarizes the Analysis of the β parameter for 50 dimensions.

In Fig. 3, we observe the behavior of the Sphere benchmark function, where the a2 parameter is remarkable in its impact on the productivity of the MA.

In Fig. 4, we evaluate the effect of the MA on the Rastrigin function. Here, we detect that a1 has a higher effect when varied.

In Fig. 5, we study the influence of the MA on the Wood function, where a2 shows a stronger influence when varied.

In the parameter impact study, it was determined that the beta parameter does not considerably affect the capabilities of the MA, and therefore, it has been excluded from the fuzzy adaptation. However, the a1 and a2 parameters exhibited a sharp impact in the algorithm's performance, so they will be selected to be part of the fuzzy adapter designs [34].

5 Designed Fuzzy Adapters

Fuzzy adapters were designed to address the challenge of stagnation in local optima. To tackle this issue, a series of sophisticated fuzzy adapters were developed to dynamically enhance diversity whenever necessary [35].

These adapters are carefully engineered to respond in real time to the absence of effective solutions, ensuring continuous adaptability and resilience.

The application of these advanced techniques has led to remarkable success, significantly surpassing the performance of the original Mayfly algorithm.

By refining its exploratory capabilities and mitigating premature convergence, these fuzzy allow the algorithm to explore intricate complex solution domains more competently and identify superior solutions [36].

5.1 Designed Fuzzy Adapter with One Input and One Output

A fuzzy adapter was planned to address the rapid convergence of MA, which tends to have low exploration and high exploitation. This adapter fuzzily adjusts the parameter a2, aiming at helping the algorithm in evading local optima, considering diversity and the number of iterations encountered.

In Fig. 6, we can observe the structure of the first fuzzy adapter, where the iteration was used as the input and the parameter a2 as the output to accelerate the convergence of the original Mayfly algorithm. The fuzzy rules are listed in Table 5. For this purpose, triangular membership functions were used:

$$f(x;a,b,c)=\begin{cases}0, & x\le a\\ \dfrac{x-a}{b-a}, & a\le x\le b\\ \dfrac{c-x}{c-b}, & b\le x\le c\\ 0, & c\le x\end{cases}, \tag{17}$$

In Equation 17, we have triangular MF where the values of a, b, and c will be provided in the Table 6 [37].

In Table 5, we can notice that there is a total of 5 fuzzy rules for this first fuzzy adapter. In these rules, the parameter a2 is manipulated.

Table 12. Performance of the MA for different mathematical functions at 20 dimensions

Function		MA	FMA(Two inputs and two outputs)	MUFMA(Fuzzy mu Mayfly)
F1	$\bar{x}$	1.177x10^{-83}	1.177x10^{-83}	2.006x10^{-84}
	S	4.617x10^{-82}	4.617x10^{-82}	7.205x10^{-84}
F2	$\bar{x}$	2.853x10^{0}	2.853x10^{0}	1.359x10^{0}
	S	2.546x10^{0}	2.546x10^{0}	1.153x10^{0}
F3	$\bar{x}$	2.982x10^{0}	2.982x10^{0}	2.980x10^{0}
	S	1.160x10^{-2}	1.160x10^{-2}	1.355x10^{-15}
F4	$\bar{x}$	1.810x10^{-2}	1.810x10^{-2}	7.401x10^{-18}
	S	5.670x10^{-2}	5.670x10^{-2}	4.053x10^{-17}
F5	$\bar{x}$	8.301x10^{3}	8.301x10^{3}	8.301x10^{3}
	S	8.300x10^{3}	8.300x10^{3}	5.550x10^{-12}
F6	$\bar{x}$	6.151x1^{-102}	6.151x10^{-102}	9.159x10^{-43}
	S	3.369x10^{-101}	3.369x10^{-101}	5.016x10^{-42}
F7	$\bar{x}$	3.060x10^{-2}	3.060x10^{-2}	4.400x10^{-3}
	S	1.641x10^{0}	1.641x10^{0}	1.219x10^{0}
F8	$\bar{x}$	2.903x10^{-15}	2.903x10^{-15}	1.567x10^{-16}
	S	1.624x10^{-7}	1.624x10^{-7}	2.899x10^{-16}
F9	$\bar{x}$	1.085x10^{-29}	1.085x10^{-29}	8.282x10^{-28}
	S	5.134x10^{-29}	5.134x10^{-29}	4.383x10^{27}
F10	$\bar{x}$	6.199x10^{-83}	6.199x10^{-83}	1.793x10^{-79}
	S	3.142x10^{-82}	3.142x10^{-82}	9.823x10^{-79}

Table 13. Performance of the MA for different mathematical functions at 50 dimensions

Function		MA	FMA(Two inputs and two outputs)	MUFMA(Fuzzy mu Mayfly)
F1	$\bar{x}$	1.177x10^{-7}	8.345x10^{-17}	8.345x10^{-17}
	S	4.520x10^{-7}	5.103x10^{-17}	5.103x10^{-17}
F2	$\bar{x}$	1.190x10^{1}	1.127x10^{1}	1.127x10^{1}
	S	3.820x10^{0}	3.792x10^{0}	3.792x10^{0}
F3	$\bar{x}$	0.000x10^{0}	3.122x10^{0}	3.122x10^{0}
	S	0.000x10^{0}	2.910x10^{-2}	2.910x10^{-2}
F4	$\bar{x}$	4.140x10^{-3}	9.024x10^{-4}	9.024x10^{-4}
	S	1.290x10^{-2}	3.800x10^{-3}	3.800x10^{-3}
F5	$\bar{x}$	3.880x10^{0}	2.075x10^{4}	2.075x10^{4}
	S	8.830x10^{-1}	2.958x10^{0}	2.958x10^{0}
F6	$\bar{x}$	5.280x10^{-49}	3.998x10^{-127}	3.998x10^{-127}
	S	1.650x10^{-48}	2.190x10^{-122}	2.190x10^{-122}
F7	$\bar{x}$	6.770x10^{1}	5.521x10^{1}	5.521x10^{1}
	S	3.987x10^{1}	2.300x10^{1}	7.918x10^{-4}
F8	$\bar{x}$	3.346x10^{-8}	7.918x10^{-4}	7.918x10^{-4}
	S	1.624x10^{-7}	2.100x10^{-3}	2.100x10^{-3}
F9	$\bar{x}$	1.713x10^{-1}	6.876x10^{1}	6.876x10^{1}
	S	8.373x10^{-2}	1.305x10^{2}	1.305x10^{2}
F10	$\bar{x}$	7.392x10^{-6}	3.694x10^{-15}	3.694x10^{-15}
	S	2.579x10^{-5}	5.609x10^{-15}	5.609x10^{-15}

5.2 Designed Fuzzy Adapter with Two Inputs and Two Outputs

This fuzzy adapter was designed to provide greater control over exploration and exploitation during each iteration of the MA.

By using iteration and diversity as inputs, the sliding diversity method is applied, allowing for a weighted average that helps determine whether there is high or low diversity in the algorithm [38].

As output parameters, a1 and a2 were chosen, as they are the main attraction constants that directly influence the algorithm's performance.

This adaptive technique secures the algorithm can dynamically adjust its behavior, promoting effective exploration when diversity is low and focusing on exploitation when diversity is high [39].

Thereby enhancing its overall efficiency and ability to escape local optima. In Table 7, we can find that a total of 9 fuzzy rules were used for this fuzzy adapter, through which we manipulate the main attraction components a1 and a2, the first is kept low and the second high if we want to increase diversity, and vice versa in the opposite case.

In Fig. 7, we can appreciate the design of the 2-input, 2-output fuzzy adapter, which was chosen based on achieving an equilibrium between exploration and exploitation in the MA. This design ensures that the algorithm can effectively adapt its behavior in response to dynamic conditions, optimizing performance throughout the search process.

5.3 Designed Fuzzy Adapter Two Inputs and Two Outputs a1 and mu

In Section 5.3, we design another fuzzy adapter, with the difference that we use the most impactful parameters in the velocity update, which are a1, and mu. The use of mutation has not been explored in depth, so we will include a range from 0.01 to 0.10 based on the typical configuration [40].

This way, we help the Mayfly algorithm escape local optima. In Fig. 8, we can find the general design of the fuzzy adapter, with iteration and diversity as inputs, and a1 and mu as outputs.

This design allows for real time adjustments to the parameters based on the algorithm's current state, ensuring improved adaptability and performance during the optimization process. Table 10 shows the parameters of the input MFs and Table 11 lists the parameters of the output MFs:

$$d = \frac{1}{n_s}\sum_{i=1}^{n_s}\sqrt{\sum_{j=1}^{n_s}(x_{ij}(t) - \overline{X}_j(t))^2}\,. \tag{18}$$

Equation 18 represents the calculation of diversity, which has been used to determine one of the inputs for the fuzzy adapters [41]. However, after calculating it, the value is normalized and a sliding calculation is performed to determine how it increases or decreases in each mathematical function.

The trapezoidal MFs are presented in Equation 19, in which the values a, b, c, and d will determine the different degrees of membership for our fuzzy adapter:

$$f(x; a, b, c, d) = \begin{cases} 0, & \text{if}(x < a) \text{ o } (x > d) \\ \dfrac{x-a}{b-a}, & \text{if } a \le x \le b \\ 1, & \text{if } b \le x \le c \\ \dfrac{d-x,}{d-c} & \text{if } c \le x \le d \end{cases}. \tag{19}$$

Equation 19 presents the trapezoidal membership functions, in which the values a, b, c, and d will determine the different degrees of membership for our fuzzy adapter.

6 Results

In this section, the results of the previously introduced fuzzy adapters are presented, exploring variations in dimensionality and employing a set of mathematical functions to compare the original method with the proposed approach, as well as with other metaheuristics or swarm intelligence algorithms.

In Table 12, we can see the performance of Mayfly and the three parameter adaptations that were implemented for 20 dimensions.

In Table 13, we can see the performance of Mayfly and the three parameter adaptations that were implemented for 50 dimensions.

In Table 14, we have statistical tests for the 10 mathematical functions used with 50 dimensions, comparing them with the fuzzy adapter that takes iteration and diversity as inputs and outputs the

Table 14. Stadistic test for different mathematical functions at 50 dimensions for FMA

Function		FMA(two inputs and two outputs)	Mayfly	
F1	x̄	8.345×10^{-17}	1.177×10^{-7}	-1.767
	S	5.103×10^{-17}	4.520×10^{-7}	
F2	x̄	1.127×10^{1}	1.190×10^{1}	-0.641
	S	3.792×10^{0}	3.820×10^{0}	
F3	x̄	3.122×10^{0}	0.000×10^{0}	0
	S	2.910×10^{-2}	0.000×10^{0}	
F4	x̄	9.024×10^{-4}	1.290×10^{-2}	-1.648
	S	3.800×10^{-3}	1.290×10^{-2}	
F5	x̄	2.075×10^{4}	8.830×10^{-1}	0
	S	2.958×10^{0}	8.830×10^{-1}	
F6	x̄	3.998×10^{-127}	5.280×10^{-49}	-1.852
	S	2.190×10^{-122}	1.650×10^{-48}	
F7	x̄	5.521×10^{1}	6.770×10^{1}	-1.647
	S	2.300×10^{1}	3.987×10^{1}	
F8	x̄	7.918×10^{-4}	3.346×10^{-8}	2.065
	S	2.100×10^{-3}	1.624×10^{-7}	
F9	x̄	6.876×10^{1}	1.713×10^{-1}	2.877
	S	1.713×10^{-1}	8.373×10^{-2}	
F10	x̄	3.694×10^{-15}	7.392×10^{-6}	-1.769
	S	7.392×10^{-6}	2.579×10^{-5}	

variables a1 and a2. Here, we can observe that a total of 5 tests are passed.

In Table 15, we can see the performance of Mayfly and statistical test for 15 different mathematical functions.

Finally, we show the results for 100 dimensions to verify that the method still behaves well with a higher complexity.

7 Discussion of the Results

When analyzing the impact of the parameters (Tables 3-5), it was confirmed how the three main position update parameters affect the performance of the Mayfly algorithm. Beta was discarded for the design of our fuzzy adapter because it did not show a significant impact.

However, the parameters a1 and a had a strong influence on the algorithm, with a2 clearly having the most significant effect on Mayfly's performance. This observation aligns with Equation 5, which indicates that a2 is multiplied by the global best solution, making it a highly important parameter.

To visualize the impact of the parameters, graphs were generated and are shown in Figures 3-7. These figures clearly demonstrate that a2 has a greater influence on the algorithm's performance when evaluated using various mathematical benchmark functions. However, in some functions, a1 showed a higher impact, which is why both parameters were used in the final fuzzy adapter.

Regarding the performance shown by Mayfly, as observed in Table 12, the algorithm performs quite well at low dimensions. Therefore, applying fuzzy parameter adaptation at lower dimensions does not significantly improve performance, as the fuzzy adapter cannot be fully utilized in such cases.

Table 13 provides a clearer view of how fuzzy parameter adaptation improves the Mayfly algorithm. It shows that the more complex the dimensionality of the problem, the more effective the fuzzy adapter becomes in handling uncertainty and, most importantly, in enhancing the robustness

Table 15. Statistic test for different mathematical functions at 50 dimensions for MUFMA

Function		FMA(two inputs and two outputs)	Mayfly	
F1	x̄	$8.211x10^{-18}$	$1.177x10^{-7}$	-1.832
	S	$5.975x10^{-17}$	$3.520x10^{-7}$	
F2	x̄	$1.086x10^{1}$	$1.190x10^{1}$	-0.449
	S	$1.197x10^{1}$	$3.820x10^{0}$	
F3	x̄	$3.026x10^{0}$	$2.987x10^{0}$	2.414
	S	$8.610x10^{-2}$	$2.030x10^{-2}$	
F4	x̄	$1.316x10^{-4}$	$4.140x10^{-3}$	-1.655
	S	$3.474x10^{-3}$	$1.280x10^{-2}$	
F5	x̄	$2.075x10^{0}$	$3.880x10^{0}$	-1.822
	S	$5.350x10^{0}$	$8.830x10^{-1}$	
F6	x̄	$9.025x10^{-140}$	$5.280x10^{-49}$	-1.752
	S	$4.943x10^{-139}$	$1.650x10^{-48}$	
F7	x̄	$5.172x10^{1}$	$6.770x10^{1}$	-1.688
	S	$3.312x10^{1}$	$3.987x10^{1}$	
F8	x̄	$1.876 x10^{-4}$	$3.346x10^{-8}$	1.855
	S	$5.536 x10^{-4}$	$1.624x10^{-7}$	
F9	x̄	$3.384x10^{1}$	$1.713x10^{-1}$	2.015
	S	$9.149x10^{1}$	$8.373x10^{-2}$	
F10	x̄	$3.542x10^{-15}$	$7.392x10^{-6}$	-1.663
	S	$2.515x10^{-15}$	$2.429x10^{-5}$	
F11	x̄	$-1.476x10^{1}$	$1.713x10^{3}$	-2.370
	S	$4.113x10^{1}$	$3.992x10^{3}$	
F12	x̄	$2.000x10^{-4}$	$9.800x10^{-1}$	-2.546
	S	$2.102x10^{0}$	$1.414x10^{-1}$	
F13	x̄	$1.256x10^{-1}$	$5.585x10^{2}$	-2.173
	x̄	$3.650x10^{-1}$	$1.407x10^{3}$	
F14	S	$1.093x10^{-1}$	$3.314x10^{-15}$	0.473
	x̄	$1.266x10^{0}$	$2.321x10^{-15}$	
F15	S	$2.146x10^{-100}$	$1.639x10^{-31}$	-1.793
	x̄	$4.267x10^{-112}$	$5.007x10^{-31}$	

and fine-tuning of the evolutionary algorithm, which could not achieve this improvement on its own.

In terms of statistical testing, we obtained a result of 5 out of 10 for the first fuzzy adapter with two inputs and two outputs. Consequently, additional tests were conducted using the fuzzy adapter that utilized μ and a2 as output parameters, with a total of 15 tests, 9 of which were successful, demonstrating its effectiveness in high-dimensional problems. These results can be found in Tables 14 and 15, respectively.

As a final test, a total of 100 dimensions were used to compare the effect of the fuzzy adapters on the Mayfly algorithm. Improvements were observed in 8 out of the 10 mathematical functions tested across the different fuzzy adapters. However, the best results were achieved using iteration and diversity as inputs and a1 and a2 as outputs.

8 Conclusions

In conclusion, the Mayfly algorithm demonstrates excellent exploration capabilities but tends to struggle with avoiding falling into local optima. From the results presented in this work, it is evident that using fuzzy parameter adaptation can be highly beneficial in addressing this issue, especially in high dimensional spaces.

In the case of low dimensions, specifically less than 20, the fuzzy adaptation does not fully leverage its potential due to the low presence of uncertainly and complexity. However, the Mayfly algorithm with fuzzy adaptation proves to be valuable for applications such as optimizing fuzzy controllers or solving high complexity problems, like flight path optimization, drone control, and similar challenges.

The results obtained with fuzzy parameter adaptation outperform those achieved with the original Mayfly algorithm. However, in low dimensional spaces, specifically with dimensions of 20 or fewer, the use of a fuzzy adapter is not advisable, as the improvement is not significant enough to justify its application.

Furthermore, it was determined that the parameter a2 has the highest repercussion on the algorithm, and maintaining its values close to one significantly enhances the overall performance of the Mayfly algorithm.

As future work, it is recommended to optimize the MFs using the Mayfly algorithm and create a fuzzy adapter with these membership values, which will later be used as a basis for developing a fuzzy adapter with type-2 fuzzy logic [42]. In addition, we could try the same approach for other algorithms [43-44].

References

1. **Dadrasajirlou, Y., Karami, H. (2024).** A survey of different Whale Optimization Algorithm applications in water engineering and management. Handbook of Whale Optimization Algorithm, pp. 613–624, Academic Press.
2. **Ghasemi, M., Zare, M., Mohammadi, S., Mirjalili, S. (2024).** Applications of whale migration algorithm in optimal power flow problems of power systems. Handbook of whale optimization algorithm, pp. 347–364, Academic Press.
3. **Zhou, J., Duan, Z., Li, Y., Deng, J., Yu, D. (2006).** PSO-based neural network optimization and its utilization in a boring machine. Journal Mater Process Technol, Vol. 178, No. 1–3, pp. 19–23. DOI: 10.1016/j.jmatprotec.2005.07.002.
4. **Yang, S., Xiong, G., Fu, X., Mirjalili, S., Mohamed, A. (2024).** Enhanced Whale optimization algorithms for parameter identification of solar photovoltaic cell models: A comparative study. Scientific Reports, Vol. 14, No. 1, pp. 16765.
5. **Majumdar, P., Mitra, S., Mirjalili, S., Bhattacharya, S. (2024).** Whale optimization algorithm-comprehensive meta analysis on hybridization, latest improvements, variants and applications for complex optimization problems. Handbook of Whale Optimization Algorithm, pp. 81–90.
6. **Sahoo, S., Reang, S., Saha, A., Chakraborty, S. (2024).** F-WOA: an improved whale optimization algorithm based on Fibonacci search principle for global optimization. Handbook of Whale Optimization Algorithm, pp. 217–233, Academic Press.
7. **Hosseini, S., Ghazi, G., Botez, R. (2023).** Application of Type One Adaptive Fuzzy Sliding Mode Control System for the Longitudinal Motion of the Cessna Citation X. AIAA 2023-3801, Estimation, Control, and Optimization. DOI: 10.2514/6.2023-3801.
8. **Zhao, L., Liu, J., Li, Y. (2024).** Application of Improved WOA in Hammerstein Parameter Resolution Problems under Advanced Mathematical Theory. Journal of Applied Mathematics, Vol. 1, pp. 5619098.
9. **Jain, M., Saihjpal, V., Singh, N., Singh, S.B. (2022).** An Overview of Variants and Advancements of PSO Algorithm. Applied Sciences (Switzerland), Vol. 12, No. 17. DOI: 10.3390/app12178392.
10. **Isaac, O., Wuraola, A., Alagbe, G. (2022).** Effect of roulette wheel selection method on Mayfly Algorithm. University of Pitesti Scientific Bulletin: Electronics and Computers Science, Vol. 22, No. 2, pp. 1-X.
11. **Nava, R., Kavitha, M., Sudarsana, G., et al. (2024).** Mayfly optimistic hyperelliptic curve cryptosystem. Front Comput Science, Vol. 6. DOI: 10.3389/fcomp.2024.1381850.
12. **Sulaiman, A.T., Bello-Salau, H., Onumanyi, A.J., et al., (2024).** A Particle Swarm and Smell Agent-Based Hybrid Algorithm for

Enhanced Optimization. Algorithms, Vol. 17, No. 2. DOI: 10.3390/a17020053.

13. **Chandrasekaran, S., Rajasekaran, V. (2024).** Energy-efficient cluster head using modified fuzzy logic with WOA and path selection using enhanced CSO in IoT-enabled smart agriculture systems. Journal of Supercomputing, Vol. 80, No. 8, pp. 11149–11190.
14. **Shokouhifar, M. (2021).** FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. Appl Soft Comput, Vol. 107. DOI: 10.1016/j.asoc.2021. 107401.
15. **Sedighizadeh, D., Masehian, E., Sedighizadeh, M., Akbaripour, H. (2021).** GEPSO: A new generalized particle swarm optimization algorithm. Math Comput Simul, Vol. 179, pp. 194–212. DOI: 10.1016/j.matcom.2020.08.013.
16. **Patil, R., Tamane, S., Rawandale, S.A., Patil, K. (2022).** A modified mayfly-SVM approach for early detection of type 2 diabetes mellitus. International Journal of Electrical and Computer Engineering, Vol. 12, No. 1, pp. 524–533. DOI: 10.11591/ijece.v12i1.
17. **Katoch, S., Chauhan, S., Kumar, V. (2021).** A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, Vol. 80, pp. 8091–8126. DOI: 10.1007/s11042-020-10139-6.
18. **Zervoudakis, K., Tsafarakis, S. (2020).** A mayfly optimization algorithm. Comput Ind Eng, Vol. 145, DOI: 10.1016/j.cie.2020. 106559.
19. **Valdez, F., Vazquez, J.C., Gaxiola, F. (2018).** Fuzzy Dynamic Parameter Adaptation in ACO and PSO for Designing Fuzzy Controllers: The Cases of Water Level and Temperature Control. Advances in Fuzzy Systems, Vol. 2018. DOI: 10.1155/2018/ 1274969.
20. **Wang, F., Zhang, H., Zhou, A. (2021).** A particle swarm optimization algorithm for mixed-variable optimization problems. Swarm Evol Comput, Vol. 60. DOI: 10.1016/j.swevo.2020.100808.
21. **Prasanth, V., Ramachandran, M., Ramu, K. (2022).** A Study on Mayfly Algorithm and Its Recent Developments. Data Analytics and Artificial Intelligence, Vol. 2, No. 2, pp. 109–116. DOI: 10.46632/daai/2/2/6.
22. **Zhao, J., Gao, Z.M. (2020).** The improved mayfly optimization algorithm with Chebyshev map. Journal of Physics: Conference Series, IOP Publishing Ltd. DOI: 10.1088/1742-6596/1684/1/012075.
23. **Gao, Z., Li, S., Zhao, J., Hu, Y. (2020).** The guaranteed convergence mayfly optimization algorithm. Proceedings 7th International Forum on Electrical Engineering and Automation, (IFEEA´20), Institute of Electrical and Electronics Engineers Inc., pp. 858–861. DOI: 10.1109/IFEEA51475.2020.00 179.
24. **Li, H., Yang, H., Zhang, L., Huang, X., Wang, H., Kang, Y. (2023).** Improved Discrete Mayfly Algorithm for Multi-objective Dynamic Network Community Detection. Journal of Frontiers of Computer Science and Technology, Vol. 17, No. 4, pp. 942–952. DOI: 10.3778/j.issn.1673-9418.2106011.
25. **Bogar, M., Shirodkar, I., Kulkarni, O., Jawade, S., Kakandikar, G. (2024).** Mayfly optimization algorithm: a review. Journal of Mechatronics and Artificial Intelligence in Engineering, Vol. 5, No. 1, pp. 17–30. DOI: 10.21595/jmai.2024.23909.
26. **Balyan, A., Ahuja, S., Kumar-Lilhore, U., Kumar-Sharma, S., Manoharan, P., et al. (2022).** A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. Sensors, Vol. 22, No. 16. DOI: 10.3390/s22165986.
27. **Zhang, H., Liu, Z., Gui, S.W., Zou, M., Wang, P.Y. (2022).** Improved mayfly algorithm based on hybrid mutation. Electron Lett, Vol. 58, No. 18, pp. 687–689. DOI: 10.1049/ell2.12568.
28. **Wang, Y. (2024).** Research and application of whale optimization algorithm. Computer Engineering & Science, Vol. 46, No. 5, pp. 881.
29. **Kadry, S., Rajinikanth, V., Koo, J., Kang, B.G. (2021).** Image multi-level-thresholding with Mayfly optimization. International Journal of Electrical and Computer Engineering, Vol. 11, No. 6, pp. 5420–5429. DOI: 10.11591/ijece.v11i6.pp5420-5429.

30. **Gao, Z., Zhao, J., Li, S.-R., Hu, Y.-R. (2020).** The improved mayfly optimization algorithm with opposition-based learning rules. Journal of Physics Conference Series, Vol. 1693, No. 1, pp. 012117. DOI: 10.1088/1742-6596/1693/1/012117.
31. **Lizárraga, E., Valdez, F., Castillo, O., Melin, P. (2025).** Roach infestation optimization algorithm with enhanced performance using automatic parameter adaptation based on fuzzy systems. DOI: 10.1007/978-3-031-88279-1_13.
32. **Xu, H., Liu, W.D., Li, L., Yao, D.J., Ma, L. (2024).** FSRW: Fuzzy logic-based whale optimization algorithm for trust-aware routing in IoT-based healthcare. Scientific Reports, Vol. 14, No. 1, pp. 16640.
33. **Al-Agamy, S.A., Mutaher Ba-Alwi, F., Mohsen, A.M. (2022).** A Fuzzy Logic for Parameter Adaptation in Ant Colony Optimization Approach. International Journal of Innovative Science and Research Technology, Vol. 7, No. 5.
34. **Sennan, S., Ramasubbareddy, S.S., Balasubramaniyam, A., Nayyar, M., Abouhawwash, Hikal, N.A. (2021).** T2FL-PSO: Type-2 Fuzzy Logic-Based Particle Swarm Optimization Algorithm Used to Maximize the Lifetime of Internet of Things. IEEE Access, Vol. 9, pp. 63966–63979. DOI: 10.1109/ACCESS.2021.306 9455.
35. **Xu, L., Song, B., Cao, M. (2021).** An improved particle swarm optimization algorithm with adaptive weighted delay velocity. Systems Science and Control Engineering, Vol. 9, No. 1, pp. 188–197. DOI: 10.1080/21642583.2021.1891153.
36. **Dinh, V., Kim, G. (2021).** Online self-calibration of multiple 2D LiDARs using line features with fuzzy adaptive covariance. IEEE sensors Journal, Vol. 21, No. 12, pp. 13714–13726. DOI: 10.1109/JSEN.2021. 3053260.
37. **Kumar, T., Kumar-Bhargava, A., Sharma, M.K., Dhiman, N., Nain, N. (2024).** Hybrid approach of type-2 fuzzy inference system and PSO in asthma disease. Clinical eHealth, Vol. 7, pp. 15–26. DOI: 10.1016/j.ceh.2024.01.001.
38. **Al Bataineh, A., Manacek S. (2022).** MLP-PSO Hybrid Algorithm for Heart Disease Prediction. Journal Pers Med, Vol. 12, No. 8. DOI: 10.3390/ jpm12081208.
39. **Yang, K., Pan, D. (2023).** An Improved Mayfly Optimization Algorithm for Type-2 Multi-Objective Integrated Process Planning and Scheduling. Mathematics, Vol. 11, No. 20. DOI: 10.3390/math11 204384.
40. **Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., Prasath, V. (2019).** Choosing mutation and crossover ratios for genetic algorithms - A review with a new dynamic approach. Information, Vol. 10, No. 12, pp. 390. DOI: 10.3390/info10120390.
41. **Cheng, S., Shi, Y., Qin, Q., Zhang, Q., Bai, R. (2014).** Population diversity maintenance in brain storm optimization algorithm. De Gruyter Open, Vol. 4, No. 2, pp. 83–97. DOI: 10.1515/jaiscr-2015-0001.
42. **Mittal, K., Jain, A., Vaisla, K., Castillo, O., Kacprzyk, J. (2020).** A comprehensive review on type 2 fuzzy logic applications: Past, present and future. Eng Appl Artif Intell, Vol. 95. DOI: 10.1016/ j.engappai.2020.103916.
43. **Castillo, O., Melin, P., Ontiveros, E., Peraza, C., Ochoa, P., Valdez F., Soria, J. (2019).** A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Engineering Applications of Artificial Intelligence, Vol. 85, pp. 666–680.
44. **Sanchez, D., Melin, P., Castillo, O. (2020).** Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition. Journal of Intelligent & Fuzzy Systems, Vol. 38, No. 3, pp. 3229–3252.

Article received on 25/02/2025; accepted on 05/05/2025.
**Corresponding author is Oscar Castillo.*