ISSN 2007-9737

Generalized Type-2 Fuzzy Logic-Enhanced Mayfly Algorithm for Robust and Adaptive Optimization

Enrique Lizarraga, Fevrier Valdez*

Tijuana Institute of Technology, TecNM,
Mexico

arcenio.lizarraga201@tectijuana.edu.mx; fevrier@tectijuana.mx

Abstract. The Mayfly Algorithm (MA) has demonstrated great potential as a metaheuristic for solving complex optimization problems. In this work, we propose the integration of generalized Type-2 fuzzy logic for dynamic parameter adaptation in MA, enabling accurate tuning under uncertain conditions. The proposed method improves convergence, robustness, and solution quality, making it suitable for a wide variety of applications, ranging from engineering and bioinformatics to systems biology and computational modeling. Comparative evaluation with mathematical benchmark functions demonstrates that the incorporation of generalized Type-2 fuzzy logic in MA outperforms traditional variants, highlighting its versatility and effectiveness in addressing diverse and complex optimization challenges.

Keywords. Generalized type-2, fuzzy logic, mayfly algorithm, uncertainty handling, adaptive optimization.

1 Introduction

The realm of optimization has long been influenced by classical methodologies valued for their mathematical rigor. A significant challenge emerges, however, when these approaches encounter complex systems characterized by nonlinearity and high dimensionality, mainly due to their dependence on initial conditions and the need for prior knowledge of the system [1]. In response to these limitations, genetic algorithms (GAs) emerged, modeled on the mechanisms of natural selection processes, facilitating for a broader exploration of the solution space. Even so, GAs were not without their problems, sometimes showing difficulties in converging efficiently [2].

These limitations served as a catalyst for the emergence of collective intelligence algorithms, which introduced a novel paradigm centered on cooperative behavior among simple computational agents, which introduced a different paradigm based on cooperation among simple agents. The resulting synergy proved capable of generating robust solutions applicable to different optimization scenarios, surpassing the performance offered by genetic algorithms in numerous scenarios [3].

The versatility of these methods has become evident in multiple engineering and applied science disciplines, even making inroads into highly specialized fields such as medicine, where they now contribute to challenges like genome reconstruction and pharmacokinetic modeling [4-5].

More recently, the incorporation of fuzzy logic systems to dynamically adapt parameter tuning has added a new level of sophistication to metaheuristics. This combination allows for a more natural management of the ambiguity and approximation inherent in real-world problems [6]. In this regard, generalized fuzzy logic Type-2 has stood out for its ability to perform remarkably flexible and robust tuning, effectively managing the critical trade-off of venturing into uncharted sections of the search space versus refining already discovered high-potential solutions [7-8].This work proposes the dynamic adaptation of the parameters of the MA using a generalized Type-2 fuzzy system, with the aim of increasing its performance and robustness in different optimization scenarios.

This article contains the following sections**:** Section 2 presents an adaptative optimization under uncertainly. Section 3 describes the Mayfly algorithm. Section 4 illustrates the importance of fuzzy systems. Section 5 details the proposed

parameter adaptation using generalized Type-2 fuzzy logic. Section 6 explains the design of the proposed fuzzy adapter. Section 7 shows the resulting performance obtained in dimensions of 50, 1000 and 2000 variables. Section 8 analyzes the behavior of the algorithm along with that of the different fuzzy adapters, and finally compares it with fire (PSO) and Firefly Algorithm (FA). Finally, Section 9 presents the conclusions drawn from the experiments conducted.

2 Adaptative Optimization under Uncertainly

In real world optimization environments, algorithms inevitably face various forms of uncertainty, such as the inherent variability of data, the changing dynamics of the problem, a lack of knowledge or epistemic uncertainty, and temporal alterations of the environment [9]. In such scenarios, traditional deterministic methods are often insufficient, as they operate under the assumption of a static, completely defined, and unambiguous landscape [10].

Faced with this limitation, adaptive optimization emerges, an approach focused not only on finding optimal solutions but also on the algorithm self-regulating its behavior based on environmental conditions [11].

This type of optimization incorporates internal adjustment, learning, and feedback mechanisms that allow for the dynamic modification of system parameters, such as inertia or learning coefficients in PSO, the exploration/exploitation strategy, and even the algorithm's structure as uncertainty evolves[12]. In particular, hybrid or bio-inspired methods demonstrate how fuzzy logic can be used to adapt these parameters in each iteration and improve performance in uncertain environments [13].

In this context, fuzzy logic is key for algorithms to handle uncertainty and imprecision gradually. Thanks to fuzzy logic, the optimizer can dynamically adjust parameters such as learning rates, exploration coefficients, and cognitive and social weightings according to environmental conditions, without relying on strict rules [14]. Achieving this level of self-regulation is particularly challenging even for hybrid algorithms based on PSO, where the combination of strategies alone does not guarantee optimal parameter tuning under dynamic uncertainty [15-16].

3 Mayfly Algorithm

The mayfly is an insect whose primary goal is to reproduce before disappearing. Inspired by its intense search for a mate, the mayfly algorithm represents an improvement on PSO, optimizing its convergence through the continuous updating of particle positions. This makes it possible faster and more efficient performs an efficient space search, facilitating the attainment of global optima. This methodology was introduced by Zervoudakis and Tsafarakis in 2020 [17].

3.1 Fundamental Principles of the Mayfly Algorithm

The MA algorithm begins its optimization process by creating two distinct subpopulations: males and females. Each individual, represented as a d-dimensional vector, encodes a prospective solution within the search space. Once generated, all individuals are evaluated using an objective or fitness function $f(x)$, which determines their effectiveness for the problem at hand [18].

3.2 Dynamic Behavior of Males

The positioning of males mayflies is influenced by their positions based on their previous positions and interactions with other nearby mayflies. The new position of the i-th mayfly at iteration t+1 is obtained by adding its current velocity to its previous position, as shown in equation (1). This mechanism allows individuals to explore the search space and adjust their trajectories according to social influence and the attraction behavior characteristic of the Mayfly model [19]:

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

The Cartesian or Euclidean distance is calculated as shown in equation (2):

$$\|x_i - X_i\| = \sqrt{\sum_{j=1}^{n}(x_{ij} - X_{ij})^2}, \tag{2}$$

where x_{ij} is the j[th] element of mayfly i and Xi corresponds to $pbest_i$ or gbest.

The velocity of the male mayfly is updated as indicated by equation (3):

$$v_{ij}^{t+1} = v_{ij}^{t} + a_1 e^{-vd_p^2}\left(\text{pbest}_{ij} - x_{ij}^{t}\right) + a_2 e^{-vd_g^2}\left(\text{gbest}_j - x_{ij}^{t}\right), \quad (3)$$

Where v_{ij}^{t} corresponds to the velocity of mayfly i in dimension j=1,…,n at time step t, $a_1$and a_2 are positive attraction constants, pbest_{ij} is the best local position , d_p and d_g are Euclidean distances between mayfly i and its best position and between mayfly i and the best position, respectively and $\text{gbest}_j$is the best global position mayfly [20].

The positions of the female mayflies are described by the position vector y_{ij}^{t} and its associated speed v_{ij}^{t} where j=1, denotes the dimension of the search space and the iteration time. The time update follows the standard relationship expressed in equation (4):

$$y_{ij}^{t+1} = y_{ij}^{t} + v_{ij}^{t}. \quad (4)$$

The velocity of female mayflies is updated based on the attraction to fitter males and a random component that promotes exploration, as shown in equation (5):

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^{t} + a_2 e^{-\beta r_{mf}^2}\left(x_{ij}^{t} - y_{ij}^{t}\right), & if\ (f(y_i) > f(x_i)) \\ v_{ij}^{t} + fl r, & if\ (f(y_i) \le f(x_i)\ ) \end{cases}, \quad (5)$$

Where v_{ij}^{t} denotes the velocity of female mayfly i in dimension j= 1, …, n at time step t,The term y_{ij}^{t} corresponds to the position of female mayfly i in dimension j at time step t, a_2 is a positive attraction constant and β is a fixed visibility coefficient, while r_{mf} is the Cartesian distance between male and female mayflies, finally, fl is a random walk coefficient, used when a female is not attracted by a male, so it flies randomly and r is a random value in the range [-1,1] [21].

3.3 Impact of Parameters

Analyzing equation (3), we observe that parameters a_1 and a_2 are fundamental components, as they determine the exploit-explore balance processes. An inappropriate configuration of these parameters can trap the algorithm int suboptimal regions or, conversely, to adopt excessively erratic and chaotic behavior [17].

According to previous studies, parameter a_2 is the most influential factor in the algorithm's overall behavior, since it directly controls the update towards the best global solution, while a_1 acts on the local adjustment [22]. Consequently, a_2 it has critical impact on the exploit-explore balancing during the convergence process [23].

4 Fuzzy System for Parameter Adaptation

This section presents the theoretical foundations of fuzzy systems and their relevance in the context of adaptive optimization.

Fuzzy logic has become established as a practical instrument for modeling the stochasticity inherent in dynamic systems and optimization processes operating in noisy or changing environments [24].

Its main advantage lies in its ability to represent uncertain or imprecise knowledge using linguistic rules, supporting a phased transition across states instead of relying on strict boundaries [25].

In this context, the representation of uncertainty ranges from Type-1 fuzzy logic, which models membership using precise degrees between 0 and 1, to Type-2 fuzzy logic, which adds an extra layer of uncertainty to better reflect the imprecision in the membership functions [26].

The following subsections present each of these fuzzy logic frameworks in detail, emphasizing their characteristics, limitations, and relevance to adaptive parameter control.

Type-1 Fuzzy Adaptation for Evolutionary Algorithm

4.1 Type-1 Fuzzy Adaptation for Evolutionary Algorithm

Type-I fuzzy systems represent the foundation of uncertainty modeling in parameter adaptation [27].

They are based on membership functions that assign a unique value to each input, allowing linguistic knowledge to be translated into interpretable computational rules; thanks to their low computational cost, these systems facilitate

the dynamic adjustment of algorithmic parameters [28-29].

In the field of evolutionary optimization, Type-I fuzzy logic has proven effective in adapting algorithm behavior in real time, responding to gradual changes in performance or population diversity [30].

Nevertheless, the efficacy of Type-1 fuzzy systems depends on a precise definition of the membership functions. In dynamic or uncertain environments, this rigidity can reduce their representational capacity and lead to less robust decisions. Even so, their simplicity and interpretability make them a solid starting point [31].

Although their limitations in the face of uncertainty have driven the development of more expressive models, such as the Type-2 fuzzy systems analyzed in the following section 4.2.

4.2 Interval Type-2 Fuzzy Systems

Interval Type-2 fuzzy systems (IT2-FLS) are a natural extension of Type-1 models, developed to improve the representation of uncertainty associated with imprecise or variable information. Unlike Type-1 systems, where each element has a single degree of membership, in IT2-FLS this degree is expressed by an interval that reflects the potential variability of membership [32].

Equation (6) mathematically represents this relationship:

$$\tilde{\mu}(x) = [\underset{\sim}{\mu}(x), \tilde{\mu}(x)], \tag{6}$$

where $\underset{\sim}{\mu}(x)$ represent the lower limits and $\tilde{\mu}(x)$ represent the upper limits of the membership function. The difference between these two values constitutes the so-called Footprint of Uncertainty (FOU), which quantifies the extent of uncertainty present in the system [33].

One of the most widely used variants is IT2-FLS, in which uncertainty is represented by upper and lower bounds on the membership functions [34].

This approach offers a suitable balance between modeling power and computational complexity, which has favored its adoption in various applications, especially in adaptive control, and evolutionary optimization, as in the case of the PSO algorithm [35-36].

4.3 Importance of Generalized Type-2 Fuzzy Logic

Generalized fuzzy logic Type-2 (GT2-FLS) overcomes the limitations of interval-based Type-2 fuzzy systems when uncertainty is not homogeneous or exhibits correlations between parameters. This generalization allows for the modeling of complete membership surfaces instead of simple intervals, capturing complex variations in uncertainty more accurately [37 - 38].

Its relevance is evident in the robustness and adaptability it offers in dynamic and non-deterministic environments. For example, implementing GT2-FLS improves parameter tuning mechanisms in system control and optimization, increasing stability and accuracy with imprecise data [39].

Although its implementation entails a higher computational cost, its flexibility and ability to handle complex uncertainty justify its use in evolutionary algorithms and adaptive control systems [40].

5 Proposed Method

With the aim of improving the Mayfly algorithm's ability to adapt to complex problems, a generalized fuzzy logic system of Type-2 is proposed for the dynamic adaptation of one of its fundamental parameters.

Unlike the interval-based Type-2 fuzzy logic approach, the generalized version allows for a more accurate representation of uncertainty and more robust handling of the variability in the optimization environment [41].

This improvement provides a more robust equilibrium between global search and local refinement phases, strengthening the algorithm's ability to avoid local optima and adapt to dynamic scenarios.

Figure 1 presents the general diagram of the proposed method, highlighting the main stages of the modified algorithm and the incorporation of the GT2FLS fuzzy system in its iterative process.

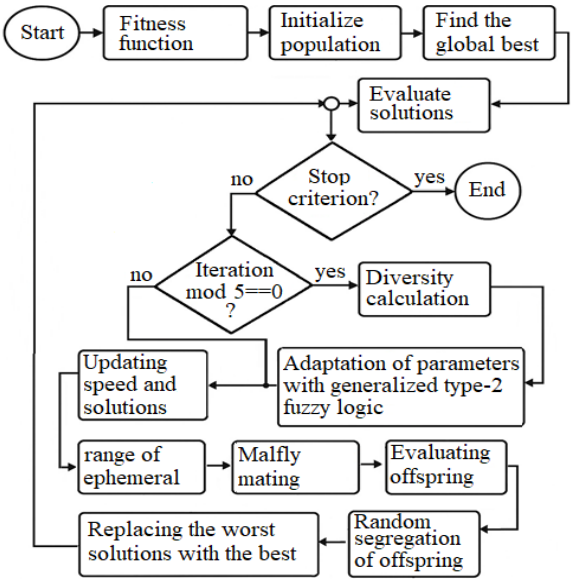

Fig. 1. General diagram of the proposed method

Figure 1 shows the operational flow of the proposed method, describing the stages of the Mayfly algorithm improved with a generalized Type-2 fuzzy system.

The process begins with defining the objective function and generating the population, followed by evaluating solutions and selecting the best overall individual. Periodically, the population diversity is calculated, and fuzzy parameter adaptation is performed, dynamically adjusting the algorithm's behavior. Finally, the solutions are updated and the least favorable ones are replaced until the stopping criterion is met.

In evolutionary algorithms, diversity measures the dispersion among candidate solutions and allows for maintaining a balance between exploration and exploitation. A diverse population favors global search, while low diversity drives local intensification; therefore, this metric reflects the influence of solutions on the evolution of the process [42-43].

In the proposed method, population diversity and the current iteration are used as inputs to the generalized Type-2 fuzzy system, which adaptively adjusts the parameters of the Mayfly algorithm according to the state of the process.

The L1 norm (Manhattan distance), a stable and computationally efficient measure, is used to calculate diversity [44-45]. Based on this value, the fuzzy system dynamically adjusts the balance between exploration and exploitation, increasing the algorithm's adaptability and performance in highly uncertain or complex scenarios. The diversity calculation is explicitly presented in equation (7):

$$\|x_i - X_i\| = \sqrt{\sum_{j=1}^{n}(x_{ij} - X_{ij})^2}, \tag{7}$$

Where n_s Denote the number of individuals solutions in the population, d is the number of decision variables or dimensions, $x_{ij}(t)$ represents the value of the j variable por the i individual at time t , $\overline{X}_j(t)$ is the average value of the j variable across the entire population at time t.

6 Design of the Generalized Type-2 Fuzzy Systems

This section presents the design of the fuzzy adapter based on a GT2-FLS, integrated into the Mayfly algorithm to adaptively optimize its internal parameters.

Unlike the interval-based Type-2 approach, the generalized system allows uncertainty to be represented by continuous membership surfaces, offering a more accurate model of the variability inherent in the evolutionary process [46].

The continuous representation of uncertainty in GT2-FLS allows for a more precise adjustment of the adaptive parameters, achieving superior performance compared to classical hybridizations or variants proposed by different authors [47-50].

6.1 General Structure of the Fuzzy System

The proposed adapter is based on a generalized Type-2 fuzzy logic system with a Mamdani architecture, designed to dynamically adjust the a_2 parameter of the Mayfly algorithm. The system has two input variables and one output.

The inputs correspond to the iteration and the population diversity, the latter calculated using the L1 norm, which measures the average dispersion of the solutions within the population. These variables allow capturing both the convergence state and the degree of exploration in the evolutionary process.

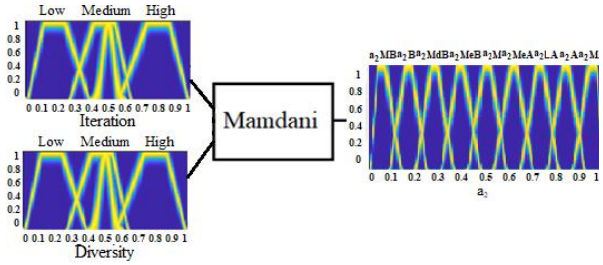

Fig. 2. General fuzzy adapter

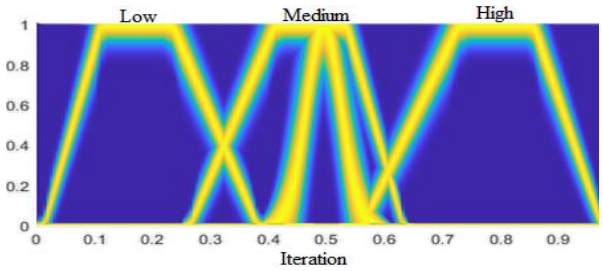

Fig. 3. Input fuzzy memberships for the iteration parameter

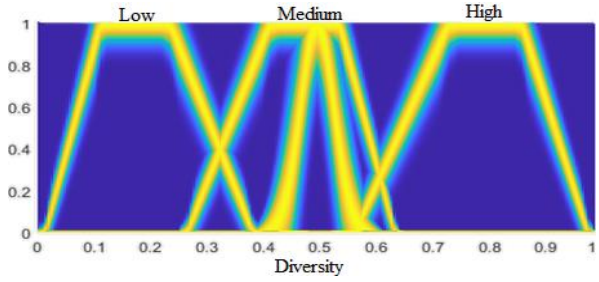

Fig. 4. Input fuzzy memberships for the diversity parameter

The output of the fuzzy system is the a_2 parameter, responsible for regulating the balance between exploration and exploitation in the algorithm [51].

Using the GT2-FLS shown in Figure 2, this parameter is adaptively adjusted based on current process conditions, enabling more precise and stable control than that achieved with static schemes or Type-1 and Type-2 interval approaches [52].

Figure 2 represents the generalized Type-2 fuzzy system, based on the Mamdani inference model, which uses trapezoidal membership functions as the primary basis for representing input and output variables. This approach allows for more accurate modeling of the uncertainty inherent in the boundaries of fuzzy sets and improves the ability to represent imprecise knowledge.

6.2 Modeling of Membership Functions

The proposed generalized fuzzy Type-2 adapter is defined by membership functions whose lower and upper bounds are modeled as bivariate functions [53]. This formulation extends the conventional interval Type-2 representation by introducing an explicit functional dependency on both the primary variable and the secondary membership degrees. Formally, this characterization is expressed by equation (8):

$$\tilde{A} = \{(x, u, \mu_{\tilde{A}}(x,u)) | x \in X, u \in U \equiv [0,1]\}, \quad (8)$$

Where $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$ the secondary universe U is assumed to be [0,1]. The membership function of T2FS it three-dimensional, with the primary variable x on the x-axis, the secondary variable u on the y-axis, and the membership grade $\mu_{\tilde{A}}$(x, u) on the z-axis.

6.2.1 Input Membership Functions

The structure of the fuzzy sets associated with the Iteration variable of the GT2FLS system is presented in Figure 3.

Figure 3 shows the fuzzy sets associated with the input variable Iteration in the GT2FLS system, defined with three linguistic labels: Low, Medium, and High.

The main membership functions are modeled using trapezoidal shapes, which are overlapped in a controlled manner to represent the progressive transitions between the different states of the evolutionary process. shown in Figure 4.

The horizontal axis represents the degree of advancement of the normalized algorithm in the range [0, 1], while the vertical axis indicates the degree of membership in each set.

Figure 4 represents the linguistic sets of the Diversity variable, also defined as Low, Medium, and High, with principal trapezoidal membership functions that describe the dispersion of the population within the evolutionary process. The horizontal axis corresponds to the normalized diversity values, and the vertical axis to the degree of membership of each value to the defined sets.

Table 1. Membership values of the input variables

Membership functions	a	b	c	d	Λ	ℓ_1	ℓ_2
Low Iteration	0	0.1	0.2	0.4	0.8	0.2	0.2
Medium Iteration	0.2	0.4	0.5	0.6	0.8	0.2	0.2
High Iteration	0.5	0.7	0.8	1	0.8	0.2	0.2
Low Diversity	0	0.1	0.2	0.4	0.8	0.2	0.2
Medium Diversity	0.2	0.4	0.5	0.6	0.8	0.2	0.2
High Diversity	0.5	0.7	0.8	1	0.8	0.2	0.2

Table 2. Membership values of the output variables

Membership functions	a	B	c	d	λ	ℓ_1	ℓ_2
a_2 Very low	1.95	1.95	1.96	1.97	0.8	0.2	0.2
a_2 Low	1.96	1.98	1.98	2.00	0.8	0.2	0.2
a_2 Moderately low	1.98	2.00	2.00	2.02	0.8	0.2	0.2
a_2 Medium low	2.01	2.02	2.03	2.04	0.8	0.2	0.2
a_2 Medium	2.03	2.04	2.05	2.06	0.8	0.2	0.2
a_2 Medium High	2.05	2.06	2.07	2.09	0.8	0.2	0.2
a_2 Slightly tall	2.07	2.09	2.09	2.11	0.8	0.2	0.2
a_2 High	2.09	2.11	2.12	2.13	0.8	0.2	0.2
a_2 Very high	2.12	2.13	2.14	2.15	0.8	0.2	0.2

Table 3. Membership values of the output variables

Rule	Iteration	Diversity	a_2
R_1	Low	Low	Very low
R_2	Low	Medium	Low
R_3	Low	High	Moderately low
R_4	Medium	Low	Medium low
R_5	Medium	Medium	Medium
R_6	Medium	High	Medium High
R_7	High	Low	Slightly tall
R_8	High	Medium	High
R_9	High	High	Muy High

This approach gives GT2-FLS a refined ability to capture imprecise variations and local fluctuations in the algorithm's behavior, maintaining a balance between interpretability and sensitivity.

To dynamic changes in process conditions. However, an excessive increase in the amplitude of the secondary Gaussian functions could excessively widen the uncertainty footprint, hindering the interpretation of fuzzy rules and reducing the clarity of the system's inference [54].

The input membership values are shown in Table 1.

Table 1 presents the membership degrees of the input variables, which were defined equally among their three linguistic terms and parameterized by the values a, b, c and d.

In the case of the parameter λ, its value was determined by exhaustive tests carried out with increments of 0.01, a procedure that was applied analogously for the parameters ℓ_1 and ℓ_2 ,whose values are presented in Table 2.

6.2.2 Output Membership Functions

Table 2 represents the membership degrees for the output membership functions, for which nine linguistic rules were defined, whose membership values were determined experimentally from 1 and 1.5. These ranges were adjusted incrementally or decreasingly in 0.1 according to the results obtained from averages of 30 executions, in order to improve the performance of the fuzzy system.

6.3 Fuzzy Rule Base

The parameter a_2 is dynamically regulated by nine fuzzy rules that automatically balance exploration and exploitation.

The fuzzy rules will be shown in Table 3 below. As shown in Table 3, in the early stages or with high diversity, a_2 is kept low to prioritize exploration of the search space.

As iterations increase or diversity decreases, the system progressively increases a_2 to intensify exploitation around promising solutions. This continuous adaptation ensures smooth transitions between search phases, optimizing the algorithm's performance throughout the evolutionary process.

Algorithm 1: Mayfly with general Type-2 fuzzy logic

Input: Objective function, Search bounds, Population size, Max iterations
Output: Best solution, Final population

```
1: Initialize male and female populations
2: Initialize GlobalBest = ∞, diversityHistory = []
3: Load GT2 fuzzy system: gt2_fis ← 'MaFuzzyt2g'
4: for it = 1 to MaxIt do
5:   norm_it ← it/MaxIt
6:   //Fuzzy adapter (main contribution)
7:   if it mod 5 == 0 then
8:      diversity ← calculate_diversity(population)
9:      norm_div←normalize(diversity,  history)
10:      a2 ← eval_gt2f([norm_it, norm_div], gt2_fis)
11:  end if
12:  // Continue with standard Mayfly operations...
13: end for
```

Algorithm 2: Diversity Calculation and Normalization

```
FUNCTION CALCULATE_DIVERSITY(population)
Input: population (matrix of size nPop × dimensions)
Output: diversity value
1: [nPop, dimensions] ← SIZE(population)
2: mean_positions ← MEAN(population, axis=0)
3: deviations ← ABS(population - mean_positions)
4: total_deviation ← SUM(deviations)
5: diversity ←total_deviation / (nPop × dimensions)
6: RETURN diversity
END FUNCTION
```

Algorithm 3: Diversity Normalization with Sliding Window

```
FUNCTION NORMALIZE_DIVERSITY(diversity, history, window)
Input: current_diversity, history_array, window_size
Output: normalized_diversity, updated_history

1: history ← APPEND(history, diversity)
2: IF LENGTH(history) > window THEN
3:    history ← history[LENGTH(history)-window:END]
4: END IF
5: minDiv ← MINIMUM(history)
6: maxDiv ← MAXIMUM(history)
7: normalized_diversity ← (diversity - minDiv) / (maxDiv - minDiv + ε)
8: RETURN normalized_diversity, history
END FUNCTION
```

6.4 Integration with the Mayfly Algorithm

Two key metrics were used to integrate the generalized Type-2 fuzzy adapter into the mayfly algorithm: normalized iterative progress and population diversity. These were processed through a fuzzy inference system that handles the inherent uncertainty in these indicators.

The mechanism is activated every 5 iterations within the main flow of the algorithm (Algorithm 1), first calculating the current population diversity (Algorithm 2), normalizing it within a consistent range (Algorithm 3), and finally evaluating the fuzzy rules to obtain the new value of a_2. This adapted parameter is immediately used in updating male speeds, directly influencing their search behavior**.**

7 Results

This part of the paper elucidates the findings from adapting parameters using generalized Type-2 fuzzy logic applied to the Mayfly algorithm (MAF2G).

The analysis considers variations in dimensionality and evaluates the performance of the proposed approach compared to three reference configurations: the interval-based Type-2 version (MAF2I), the Type-1 version (MAF1), and the original Mayfly without fuzzy adaptation. Comparisons with other representative swarm intelligence metaheuristics, such as PSO and the FA, are also included.

The experimental tests were performed using a population of 40 agents and a maximum of 2000 iterations, under homogeneous evaluation conditions to ensure the comparative validity of the results.

The mathematical functions used for performance evaluation are presented in Table 4, which allow for quantifying the accuracy, stability, and convergence capacity of the proposed approach compared to other methods.

Table 4 shows the abbreviations of mathematical functions in the left column, while their full names are indicated in the right column.

The comparative analysis of fuzzy adapters was initiated considering a set of 50 dimensions. The results obtained are presented in Table 5.

Table 5 shows the analysis of the different fuzzy adapters. MAF2G stands out from the other two, showing better performance in half of the cases. However, the difference with MAF2I is less evident when working with 50 dimensions.

Table 4. Benchmark functions.

Function	Name
F1	Sphere
F2	Rastrigin
F3	Griewank
F4	Powell
F5	Rosenbrock
F6	Alpine
F7	Zakharov
F8	Sum Squares
F9	Wood
F10	Dejong1
F11	Levy
F12	Dixon Price
F13	Qing
F14	Hyper ellipsoid

Table 5. Comparison of the performance of fuzzy adapters in 50 dimensions

Functions		MAF1	MAF2I	MAF2G
F1	$\bar{x}$	$3.151x10^{-17}$	$8.868x10^{-18}$	**$2.658x10^{-19}$**
	S	$5.975x10^{-17}$	$8.043x10^{-18}$	$3.275x10^{-19}$
F2	$\bar{x}$	**$1.086x10^{1}$**	$1.636x10^{1}$	$1.540x10^{1}$
	S	$1.197x10^{1}$	$6.484x10^{0}$	$7.334x10^{0}$
F3	$\bar{x}$	**$1.316x10^{-4}$**	$6.318x10^{-3}$	$4.793x10^{-3}$
	S	$3.474x10^{-3}$	$7.901x10^{-3}$	$4.793x10^{-3}$
F4	$\bar{x}$	$9.025x10^{-19}$	**$1.154x10^{-109}$**	$1.203x10^{-33}$
	S	$4.943x10^{-19}$	$6.152x10^{-109}$	$6.589x10^{-33}$
F5	$\bar{x}$	$5.172x10^{1}$	**$4.801x10^{1}$**	$5.577x10^{1}$
	S	$3.312x10^{1}$	$2.015x10^{1}$	$2.795x10^{1}$
F6	$\bar{x}$	$1.876x10^{-4}$	**$2.678x10^{-10}$**	$1.857x10^{-9}$
	S	$5.536x10^{-4}$	$2.842x10^{-10}$	$5.278x10^{-9}$
F7	$\bar{x}$	$3.384x10^{-1}$	**$7.031x10^{-1}$**	$2.269x10^{1}$
	S	$9.149x10^{-1}$	$1.450x10^{-2}$	$1.241x10^{2}$
F8	$\bar{x}$	$3.542x10^{-15}$	$3.809x10^{-17}$	**$8.161x10^{-18}$**
	S	$2.515x10^{-15}$	$8.202x10^{-17}$	$1.185x10^{-17}$
F9	$\bar{x}$	$2.022x10^{-1}$	**$3.599x10^{-2}$**	$1.401x10^{-1}$
	S	$1.230x10^{-1}$	$1.971x10^{-1}$	$6.510x10^{-1}$
F10	$\bar{x}$	$2.655x10^{-16}$	$1.360x10^{-17}$	**$2.629x10^{-19}$**
	S	$8.008x10^{-16}$	$3.789x10^{-17}$	$5.125x10^{-19}$

Table 6. Comparison of the performance of PSO, FA and MA2FG in 50 dimensions

Functions	PSO	FA	MA2FG	
F1	□	$1.177x10^{-3}$	**$5.383x10^{-36}$**	$2.658x10^{-19}$
	S	$4.571x10^{-3}$	$7.668x10^{-37}$	$3.275x10^{-19}$
F2	□	$4.653x10^{1}$	$1.182x10^{2}$	**$1.540x10^{1}$**
	S	$1.182x10^{1}$	$3.210x10^{1}$	$7.334x10^{0}$
F3	□	**$3.579x10^{-5}$**	$7.396x10^{-3}$	$4.793x10^{-3}$
	S	$1.880x10^{-4}$	$1.076x10^{-2}$	$4.793x10^{-3}$
F4	□	$4.725x10^{-9}$	$4.251x10^{-12}$	**$1.203x10^{-33}$**
	S	$3.079x10^{-9}$	$1.276x10^{-11}$	$6.589x10^{-33}$
F5	□	**$4.338x10^{1}$**	$3.366x10^{1}$	$5.577x10^{1}$
	S	$1.056x10^{-1}$	$1.637x10^{1}$	$2.795x10^{1}$
F6	□	$1.038x10^{-1}$	**$2.829x10^{-10}$**	$1.857x10^{-9}$
	S	$8.523x10^{-2}$	$1.576x10^{-10}$	$5.278x10^{-9}$
F7	□	**$5.007x10^{-2}$**	$7.041x10^{-2}$	$2.269x10^{1}$
	S	$5.401x10^{-2}$	$2.246x10^{-21}$	$1.241x10^{2}$
F8	□	$5.158x10^{-3}$	**$1.259x10^{-34}$**	$8.161x10^{-18}$
	S	$1.365x10^{-2}$	$1.808x10^{-35}$	$1.185x10^{-17}$
F9	□	**$7.053x10^{-6}$**	$1.200x10^{0}$	$1.401x10^{-1}$
	S	$8.006x10^{-6}$	$2.420x10^{0}$	$6.510x10^{-1}$
F10	□	$6.624x10^{-4}$	$5.368x10^{-16}$	**$2.629x10^{-19}$**
	S	$2.954x10^{-3}$	$6.037x10^{-17}$	$5.125x10^{-19}$
F11	□	$5.888x10^{-1}$	$1.181x10^{0}$	**$4.031x10^{-1}$**
	S	$2.840x10^{-1}$	$9.207x10^{-1}$	$8.707x10^{-1}$
F12	□	$1.490x10^{0}$	$6.666x10^{-1}$	**$4.562x10^{-1}$**
	S	$1.983x10^{0}$	$1.797x10^{-16}$	$3.011x10^{-4}$
F13	□	$5.525x10^{3}$	**$6.778x10^{-28}$**	$9.040x10^{-4}$
	S	$1.910x10^{-2}$	$9.361x10^{-29}$	$2.581x10^{-3}$
F14	□	$1.114x10^{-3}$	$4.2949x10^{-16}$	**$1.223x10^{-17}$**
	S	$5.747x10^{-3}$	$3.8026x10^{-16}$	$1.502x10^{-17}$

Table 6 presents the results of the PSO and FA algorithms at 50 dimensions.

Table 6 shows a comparison with other metaheuristics, PSO and FA. In this case, 50 dimensions were used, and although by a small margin, MAF2G manages to stand out against both algorithms.

To highlight the relevance of applied the generalized Type-2 fuzzy system, the dimensionality of the problem was increased to enhance its complexity and re-evaluate the performance of the fuzzy adapters.

Table 7 summarizes the results obtained for 100 dimensions.

Table 7 shows the superiority of the MAF2G adapter over the MAF2I and MAF1

This advantage is attributed to its greater capacity to handle highly complex problems.

In this case, with 1000 dimensions, MAF2G achieved the best performance in 8 of the 14 mathematical functions, followed by MAF1 with 4 out of 14, while MAF2I only excelled in 2 of the 14 functions evaluated.

Table 7. Comparison of the performance of fuzzy adapters in 1000 dimensions

Functions		MAF1	MAF2I	MAF2G
F1	□	$9.893x10^{1}$	**$9.545x10^{1}$**	$8.3376x10^{1}$
	S	$1.150x10^{1}$	$1.414x10^{0}$	$9.1463x10^{0}$
F2	□	**$1.715x10^{3}$**	$1.871x10^{3}$	$1.729x10^{3}$
	S	$1.051x10^{2}$	$1.351x10^{2}$	$1.002x10^{2}$
F3	□	$9.353x10^{3}$	$9.394x10^{3}$	**$9.292x10^{3}$**
	S	$2.473x10^{2}$	$2.039x10^{2}$	$2.033x10^{2}$
F4	□	$5.285x10^{-21}$	$1.419x10^{-33}$	**$3.610x10^{-35}$**
	S	$1.795x10^{-20}$	$7.774x10^{-33}$	$1.942x10^{-34}$
F5	□	$1.174x10^{5}$	**$8.095x10^{4}$**	$8.111x10^{4}$
	S	$1.878x10^{4}$	$1.232x10^{4}$	$1.157x10^{4}$
F6	□	$1.508x10^{2}$	$1.459x10^{2}$	**$1.456x10^{2}$**
	S	$1.810x10^{1}$	$1.492x10^{1}$	$1.421x10^{1}$
F7	□	**$2.326x10^{4}$**	$2.429x10^{4}$	$2.3245x10^{4}$
	S	$3.119x10^{3}$	$3.076x10^{3}$	$2.6702x10^{3}$
F8	□	$5.497x10^{4}$	$4.817x10^{4}$	**$4.681x10^{4}$**
	S	$9.323x10^{3}$	$5.983x10^{4}$	$4.886x10^{3}$
F9	□	$6.619x10^{-31}$	$2.568x10^{-30}$	**$5.620x10^{-31}$**
	S	$2.060x10^{-30}$	$4.283x10^{-30}$	$9.542x10^{-31}$
F10	□	**$3.955x10^{1}$**	$4.149x10^{1}$	$3.9494x10^{1}$
	S	$4.019x10^{0}$	$3.701x10^{0}$	$2.9701x10^{0}$
F11	□	**$2.382x10^{2}$**	$2.431x10^{2}$	$2.400x10^{2}$
	S	$2.438x10^{1}$	$3.441x10^{1}$	$3.062x10^{1}$
F12	□	$2.577x10^{5}$	$2.010x10^{5}$	**$1.945x10^{5}$**
	S	$3.853x10^{4}$	$3.614x10^{4}$	$2.917x10^{4}$
F13	□	$1.297x10^{12}$	$1.300x10^{12}$	**$1.256x10^{12}$**
	S	$6.155x10^{10}$	$7.580x10^{11}$	$7.804x10^{10}$
F14	□	$3.102x10^{6}$	$2.474x10^{6}$	**$2.450x10^{6}$**
	S	$8.036x10^{5}$	$4.423x10^{5}$	$4.097x10^{5}$

Table 8 presents the comparison with PSO and FA for 1000 dimensions.

Table 8 shows that MAF2G stands out in terms of performance compared to the other metaheuristics evaluated. In particular, MAF2G obtains the top results in 10 of the 14 test functions.

Table 9 presents the results obtained for the different fuzzy adapters evaluated on 2000-dimensional problems, with the aim of examining their behavior and generalizability in highly complex environments.

This analysis reveals how each adapter maintains the algorithm's stability as the search space increases exponentially, highlighting differences in their efficiency, accuracy, and adaptability to the growing difficulty of multidimensional optimization.

Table 9 shows that MAF2G performs better in 11 of the 14 mathematical functions evaluated, reflecting its greater optimization capacity and robustness in high-dimensional scenarios.

In contrast, the MAF2I adapter outperforms the other methods in only 2 functions, while MAF1

Table 8. Comparison of the performance of PSO, FA and MAF2G in 1000 dimensions

Functions		PSO	FA	MAF2G
F1	$\bar{x}$	$9.220x10^{2}$	$2.767x10^{2}$	**$8.337x10^{1}$**
	S	$1.114x10^{2}$	$4.661x10^{1}$	$9.146x10^{0}$
F2	$\bar{x}$	$8.669x10^{3}$	$6.8530x10^{3}$	**$1.729x10^{3}$**
	S	$6.094x10^{2}$	$2.556x10^{2}$	$1.002x10^{2}$
F3	$\bar{x}$	**$3.945x10^{2}$**	$6.292x10^{4}$	$9.292x10^{3}$
	S	$4.627x10^{1}$	$2.049x10^{2}$	$2.033x10^{2}$
F4	$\bar{x}$	**$0.000x10^{0}$**	**$0.000x10^{0}$**	$3.610x10^{-35}$
	S	$0.000x10^{0}$	$0.000x10^{0}$	$1.942x10^{-34}$
F5	$\bar{x}$	$3.903x10^{7}$	$1.485x10^{7}$	**$8.111x10^{4}$**
	S	$8.442x10^{6}$	$2.565x10^{6}$	$1.157x10^{4}$
F6	$\bar{x}$	$5.907x10^{2}$	$1.355x10^{3}$	**$1.456x10^{2}$**
	S	$4.192x10^{1}$	$2.109x10^{2}$	$1.421x10^{1}$
F7	$\bar{x}$	**$1.624x10^{4}$**	$7.313x10^{4}$	$2.3245x10^{4}$
	S	$7.313x10^{2}$	$2.371x10^{3}$	$2.6702x10^{3}$
F8	$\bar{x}$	$4.033x10^{5}$	$1.2884x10^{6}$	**$4.681x10^{4}$**
	S	$3.149x10^{4}$	$2.196x10^{5}$	$4.886x10^{3}$
F9	$\bar{x}$	$1.000x10^{-6}$	$5.137x10^{0}$	**$5.620x10^{-31}$**
	S	$1.000x10^{-6}$	$4.262x10^{0}$	$9.542x10^{-31}$
F10	$\bar{x}$	$1.107x10^{3}$	$6.645x10^{2}$	**$3.9494x10^{1}$**
	S	$1.897x10^{2}$	$5.444x10^{1}$	$2.9701x10^{0}$
F11	$\bar{x}$	$6.195x10^{2}$	$4.488x10^{2}$	**$2.382x10^{2}$**
	S	$6.436x10^{1}$	$5.075x10^{1}$	$2.438x10^{1}$
F12	$\bar{x}$	$9.007x10^{6}$	$3.509x10^{6}$	**$2.577x10^{5}$**
	S	$2.078x10^{6}$	$8.664x10^{5}$	$3.853x10^{4}$
F13	$\bar{x}$	$5.248x10^{9}$	**$3.094x10^{8}$**	$1.297x10^{12}$
	S	$2.026x10^{5}$	$3.951x10^{2}$	$6.155x10^{10}$
F14	$\bar{x}$	$7.433x10^{7}$	$4.228x10^{7}$	**$3.102x10^{6}$**
	S	$2.851x10^{7}$	$2.823x10^{7}$	$8.036x10^{5}$

excels in just one function, demonstrating more limited performance.

These results confirm the utility of the MAF2G focus in maintaining the algorithm's stability and accuracy even when the problem's complexity increases significantly.

The table 10 presents the performance of MAF2G versus PSO and FA in 2000 dimensions

Table 10 shows the results obtained by the PSO, FA, and MAF2G algorithms on 2000-dimensional problems.

MAF2G consistently demonstrates superior performance, excelling in 10 of the 14 mathematical functions evaluated. The FA algorithm achieves the best results in 3 functions, while PSO only excels in one.

Table 11 presents a statistical comparison between the MAF2G and MA algorithms, performed using z-tests. This analysis allows us to determine whether the observed differences in the performance of both algorithms are statistically significant, enabling a more systematic

Table 9. Comparison of the performance of fuzzy adapters in 2000 dimensions

Functions		MAF1	MAF2I	MAF2G
F1	$\bar{x}$	$6.792x10^{2}$	$6.667x10^{2}$	**$5.622x10^{2}$**
	S	$4.694x10^{1}$	$4.103x10^{1}$	$3.904x10^{1}$
F2	$\bar{x}$	**$6.262x10^{3}$**	$6.461x10^{3}$	$6.797x10^{3}$
	S	$2.626x10^{2}$	$2.556x10^{2}$	$2.787x10^{2}$
F3	$\bar{x}$	$2.509x10^{4}$	$2.511x10^{4}$	**$2.507x10^{4}$**
	S	$4.135x10^{2}$	$4.377x10^{2}$	$4.459x10^{2}$
F4	$\bar{x}$	$1.191x10^{-20}$	**$1.402x10^{-106}$**	$3.836x10^{-28}$
	S	$3.840x10^{-20}$	$7.641x10^{-106}$	$2.1012x10^{-27}$
F5	$\bar{x}$	$4.677x10^{6}$	$4.600x10^{6}$	**$2.937x10^{6}$**
	S	$1.646x10^{6}$	$1.732x10^{6}$	$1.185x10^{6}$
F6	$\bar{x}$	$6.175x10^{2}$	$6.209x10^{2}$	**$6.115x10^{2}$**
	S	$3.332x10^{1}$	$5.297x10^{1}$	$4.344x10^{1}$
F7	$\bar{x}$	$6.240x10^{4}$	$6.226x10^{4}$	**$6.150x10^{4}$**
	S	$5.897x10^{3}$	$6.852x10^{3}$	$6.444x10^{3}$
F8	$\bar{x}$	$7.135x10^{5}$	$7.020x10^{5}$	**$5.906x10^{5}$**
	S	$8.413x10^{4}$	$6.859x10^{4}$	$4.685x10^{4}$
F9	$\bar{x}$	$3.629x10^{-1}$	**$7.979x10^{-31}$**	$4.449x10^{-1}$
	S	$1.510x10^{0}$	$1.370x10^{-30}$	$1.332x10^{0}$
F10	$\bar{x}$	$3.013x10^{2}$	$3.073x10^{2}$	**$2.919x10^{2}$**
	S	$2.698x10^{1}$	$3.050x10^{1}$	$1.914x10^{1}$
F11	$\bar{x}$	$6.379x10^{2}$	$6.717x10^{2}$	**$6.034x10^{2}$**
	S	$6.834x10^{1}$	$8.748x10^{1}$	$6.195x10^{1}$
F12	$\bar{x}$	$6.712x10^{6}$	$5.948x10^{6}$	**$5.067x10^{6}$**
	S	$1.450x10^{6}$	$1.007x10^{6}$	$5.581x10^{5}$
F13	$\bar{x}$	$4.258x10^{12}$	$4.209x10^{12}$	**$4.078x10^{12}$**
	S	$1.592x10^{11}$	$2.097x10^{11}$	$2.115x10^{11}$
F14	$\bar{x}$	$7.298x10^{7}$	$6.470x10^{7}$	**$5.554x10^{7}$**
	S	$1.930x10^{7}$	$1.471x10^{7}$	$1.191x10^{7}$

assessment of the superiority and consistency of the MAF2G model compared to its counterpart.

Table 12 shows the results of the z-test applied to the 50-dimensional case, demonstrating that MAF2G significantly outperforms the classical MA in 9 of the 14 mathematical functions evaluated.

This analysis confirms that the performance differences between the two algorithms are statistically significant, supporting the effectiveness, stability, and generalizability of the MAF2G model.

Taken together, these results evidence that the proposed approach maintains competitive advantage even in scenarios of intermediate complexity, solidifying its advantage over the traditional method.

Table 12 presents the results of the Z test performed for problems with 1000 dimensions,

Table 10. Performance of MAF2G, PSO and FA at 2000 dimensions

Functions		PSO	FA	MAF2G
F1	$\bar{x}$	$3.490x10^{3}$	$2.727x10^{3}$	$\mathbf{5.622x10^{2}}$
	S	$3.864x10^{2}$	$3.366x10^{2}$	$3.904x10^{1}$
F2	$\bar{x}$	$2.126x10^{4}$	$1.537x10^{4}$	$\mathbf{6.797x10^{3}}$
	S	$1.224x10^{3}$	$2.262x10^{2}$	$2.787x10^{2}$
F3	$\bar{x}$	$1.841x10^{0}$	$\mathbf{1.646x10^{0}}$	$2.5077x10^{4}$
	S	$7.790x10^{-2}$	$2.700x10^{-2}$	$4.4593x10^{2}$
F4	$\bar{x}$	$\mathbf{0.000x10^{0}}$	$4.900x10^{-13}$	$3.836x10^{-28}$
	S	$0.000x10^{0}$	$2.660x10^{-13}$	$2.101x10^{-27}$
F5	$\bar{x}$	$3.462x10^{6}$	$3.252x10^{6}$	$\mathbf{2.937x10^{6}}$
	S	$4.931x10^{5}$	$7.529x10^{4}$	$1.185x10^{6}$
F6	$\bar{x}$	$1.496x10^{3}$	$7.575x10^{2}$	$\mathbf{6.115x10^{2}}$
	S	$7.487x10^{1}$	$1.020x10^{1}$	$4.344x10^{1}$
F7	$\bar{x}$	$6.362x10^{4}$	$5.455x10^{4}$	$\mathbf{6.150x10^{4}}$
	S	$9.070x10^{3}$	$1.141x10^{2}$	$6.444x10^{3}$
F8	$\bar{x}$	$3.075x10^{6}$	$2.308x10^{6}$	$\mathbf{5.906x10^{5}}$
	S	$2.464x10^{5}$	$1.764x10^{4}$	$4.685x10^{4}$
F9	$\bar{x}$	$1.000x10^{-6}$	$\mathbf{0.000x10^{0}}$	$4.449x10^{-1}$
	S	$3.000x10^{-6}$	$0.000x10^{0}$	$1.332x10^{0}$
F10	$\bar{x}$	$3.634x10^{3}$	$2.691x10^{3}$	$\mathbf{2.919x10^{2}}$
	S	$4.254x10^{2}$	$6.530x10^{1}$	$1.914x10^{1}$
F11	$\bar{x}$	$1.909x10^{3}$	$1.407x10^{3}$	$\mathbf{6.034x10^{2}}$
	S	$2.496x10^{2}$	$6.400x10^{0}$	$6.195x10^{1}$
F12	$\bar{x}$	$9.697x10^{7}$	$2.533x10^{9}$	$\mathbf{5.067x10^{6}}$
	S	$2.517x10^{7}$	$1.451x10^{8}$	$5.581x10^{5}$
F13	$\bar{x}$	$2.533x10^{9}$	$\mathbf{2.301x10^{7}}$	$4.078x10^{12}$
	S	$1.451x10^{9}$	$2.798x10^{4}$	$2.115x10^{11}$
F14	$\bar{x}$	$2.823x10^{8}$	$2.533x10^{9}$	$\mathbf{5.554x10^{7}}$
	S	$1.452x10^{7}$	$1.451x10^{9}$	$1.191x10^{7}$

providing a statistical comparison between the evaluated algorithms.

Table 12 shows the results of the Z-tests performed on 1000-dimensional problems, comparing the performance of MAF2G against the classical MA.

It can be seen that, as the complexity of the search space increases, MAF2G stands out more clearly, surpassing the statistical test in a total of 11 mathematical functions, thus demonstrating its robustness and effectiveness compared to the traditional method.

Table 13 presents the results of the Z statistical tests for 2000 dimensional problems, rigorously comparing the performance of MAF2G against MA.

Table 14 shows that the Z-test results are promising even in high-dimensionality scenarios, evaluating problems with 2000 dimensions.

In particular, MAF2G maintains superior performance in 11 of the 14 mathematical

Table 11. Z test with MA for 50 dimensions

Functions		MAF2G	MA	Z
F1	$\bar{x}$	$2.658x10^{-19}$	$1.177x10^{-7}$	**-1.831**
	S	$3.275x10^{-19}$	$3.520x10^{-7}$	
F2	$\bar{x}$	$1.540x10^{1}$	$1.190x10^{1}$	2.318
	S	$7.334x10^{0}$	$3.820x10^{0}$	
F3	$\bar{x}$	$2.293x10^{-4}$	$4.140x10^{-3}$	**-1.664**
	S	$1.320x10^{-3}$	$1.280x10^{-2}$	
F4	$\bar{x}$	$1.203x10^{-33}$	$5.280x10^{-49}$	1.000
	S	$6.589x10^{-33}$	$1.650x10^{-48}$	
F5	$\bar{x}$	$5.277x10^{1}$	$6.770x10^{1}$	-**1.679**
	S	$2.795x10^{1}$	$3.987x10^{1}$	
F6	$\bar{x}$	$1.857x10^{-9}$	$6.563x10^{-6}$	**-1.926**
	S	$5.278x10^{-9}$	$1.865x10^{-5}$	
F7	$\bar{x}$	$2.269x10^{1}$	$8.946x10^{-1}$	0.962
	S	$1.241x10^{2}$	$1.541x10^{-2}$	
F8	$\bar{x}$	$8.161x10^{-18}$	$7.392x10^{-6}$	**-1.666**
	S	$1.185x10^{-17}$	$2.429x10^{-5}$	
F9	$\bar{x}$	$1.401x10^{-1}$	$2.366x10^{-32}$	1.178
	S	$6.510x10^{-1}$	$1.296x10^{-31}$	
F10	$\bar{x}$	$2.629x10^{-19}$	$2.114x10^{-14}$	**-8.975**
	S	$5.125x10^{-19}$	$1.290x10^{-14}$	
F11	$\bar{x}$	$4.031x10^{-1}$	$1.850x10^{0}$	**-4.170**
	S	$8.707x10^{-1}$	$1.689x10^{0}$	
F12	$\bar{x}$	$6.666x10^{-1}$	$1.257x10^{0}$	**-2.014**
	S	$3.011x10^{-4}$	$1.605x10^{0}$	
F13	$\bar{x}$	$9.040x10^{-4}$	$2.647x10^{-2}$	-1.306
	S	$2.581x10^{-3}$	$1.102x10^{-1}$	
F14	$\bar{x}$	$1.223x10^{-17}$	$1.269x10^{-16}$	**-1.856**
	S	$1.502x10^{-17}$	$3.256x10^{-16}$	

functions, demonstrating that the observed improvements are not due to chance but are statistically significant.

These results support the advantage of using a generalized Type-2 adapter, demonstrating its ability to maintain the algorithm's robustness, stability, and effectiveness in the face of increasing problem complexity, and validating its superiority over classical MA.

8 Discussion of results

The analysis of the results shows consistent performance behavior among the fuzzy adapters. MAF2G emerged as the most efficient adapter across all evaluated dimensionalities, and its advantage became more pronounced as the problem's complexity increased.

In 50 dimensions, it outperformed the other adapters in 7 out of 14 functions, a figure that

Table 12. Z test with MA for 1000 dimensions

Functions		MAF2G	MA	Z
F1	$\bar{x}$	$8.3376x10^{1}$	$1.478x10^{2}$	**-19.599**
	S	$9.1463x10^{0}$	$1.551x10^{1}$	
F2	$\bar{x}$	$1.729x10^{3}$	$1.778x10^{3}$	**-1.879**
	S	$1.002x10^{2}$	$1.017x10^{2}$	
F3	$\bar{x}$	$9.292x10^{3}$	$9.396x10^{3}$	-1.471
	S	$2.033x10^{2}$	$3.293x10^{2}$	
F4	$\bar{x}$	$3.610x10^{-35}$	$9.571x10^{-18}$	**-14.020**
	S	$1.942x10^{-34}$	$3.739x10^{-18}$	
F5	$\bar{x}$	$8.111x10^{4}$	$2.209x10^{5}$	**-20.449**
	S	$1.157x10^{4}$	$3.561x10^{4}$	
F6	$\bar{x}$	$1.456x10^{2}$	$1.659x10^{2}$	**-4.640**
	S	$1.421x10^{1}$	$2.080x10^{1}$	
F7	$\bar{x}$	$2.324x10^{4}$	$2.549x10^{4}$	**-3.4169**
	S	$2.670x10^{3}$	$3.597x10^{3}$	
F8	$\bar{x}$	$4.681x10^{4}$	$7.533x10^{4}$	**-13.434**
	S	$4.886x10^{3}$	$1.056x10^{4}$	
F9	$\bar{x}$	$5.620x10^{-31}$	$3.845x10^{-31}$	0.8579
	S	$9.542x10^{-31}$	$6.114x10^{-31}$	
F10	$\bar{x}$	$3.9494x10^{1}$	$5.183x10^{1}$	**-10.732**
	S	$2.9701x10^{0}$	$5.551x10^{0}$	
F11	$\bar{x}$	$2.400x10^{2}$	$2.444x10^{2}$	-0.557
	S	$3.062x10^{1}$	$3.047x10^{1}$	
F12	$\bar{x}$	$1.945x10^{5}$	$4.046x10^{5}$	**-16.750**
	S	$2.917x10^{4}$	$6.220x10^{4}$	
F13	$\bar{x}$	$1.256x10^{12}$	$1.307x10^{12}$	**-2.729**
	S	$7.804x10^{10}$	$6.619x10^{10}$	
F14	$\bar{x}$	$2.450x10^{6}$	$3.277x10^{6}$	**-7.030**
	S	$4.097x10^{5}$	$4.973x10^{5}$	

increased to 8 out of 14 in 1000 dimensions and reached 11 out of 14 in 2000 dimensions.

This upward trend suggests that the generalized Type-2 fuzzy system incorporated in MAF2G offers a better capacity for handling uncertainty and complexity in high-dimensional spaces. The greater flexibility provided by this type of logic appears to favor a more suitable balance between exploration and exploitation, resulting in more stable performance as dimensionality increases.

When measure to established algorithms such as PSO and FA, the results reinforce this observation. MAF2G achieved notable improvements in highly multimodal functions with complex gradients, such as Rosenbrock (F5) and Dixon Price (F12).

This suggests that the proposed adapter not only maintains good performance in unimodal functions but also retains its effectiveness in scenarios of greater difficulty and variability.

Finally, statistical tests (Z-tests) confirm that the noted accuracy boosts are not accidental but represent consistent and statistically significant advantages of the proposed method. Taken together, the results reflect that MAF2G possesses remarkable scalability, making it a robust alternative for tackling complex optimization problems where other methods often exhibit limitations.

9 Conclusions

This research provides compelling evidence that the proposed algorithm MAF2G, which incorporates an adaptation mechanism based on generalized fuzzy logic Type-2, consistently outperforms the compared algorithms MA, PSO, FA, and even their variants with fuzzy logic Type-1 and interval Type-2, particularly as the problem's dimensionality increases.

In 50, 1000, and 2000-dimensional scenarios, MAF2G not only maintained its competitiveness but also significantly increased its advantage, achieving the best performance in a greater number of functions as the complexity of the search space grew.

This behavior confirms that the use of generalized fuzzy logic Type-2 is especially advantageous in high-dimensional problems, where uncertainty, interactions between variables, and the difficulty of adjusting parameters become critical.

Unlike algorithms with limited or rigid adaptation mechanisms, MAF2G effectively manages these uncertainties through a fuzzy system that flexibly adjusts the parameters, improving the management of exploration and exploitation.

For future research it is proposed to explore the application of this algorithm in complex problems in the medical field, such as parameter optimization for genome reconstruction and prediction of kinetic curves in pharmacological studies [55-57].

Furthermore, recent studies have shown that the integration of fuzzy adapters can significantly improve the performance of evolutionary algorithms in highly complex scenarios, supporting the relevance of extending this line of research [58- 60].

References

1. **Katoch, S., Chauhan, S., Kumar, V. (2021).** A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
2. **Mirjalili, S., Song, J., Sadiq, A., Faris, H. (2020).** Genetic algorithm: Theory, literature review, and application in image reconstruction. In Studies in Computational Intelligence (Vol. 811, pp. 69–85). Springer. https://doi.org/10.1007/978-3-030-12127-3_5
3. **Papazoglou, G., Biskas, P. (2023).** Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem. Energies. https://doi.org/10.3390/en16031152
4. **Kim, S., Hooker, A., Shi, Y., Kim, G., Wong, W. (2021).** Metaheuristics for pharmacometrics. CPT: Pharmacometrics and Systems Pharmacology, 10(11), 1297–1309. American Society for Clinical Pharmacology and Therapeutics. https://doi.org/10.1002/psp4.12714
5. **Shamami, M., Teimourpour, B., Sharifi, F. (2025).** A hybrid computational intelligence framework with metaheuristic optimization for drug-drug interaction prediction. https://doi.org/10.48550/arXiv.2510.09668
6. **Patel, H., Shah, V. (2022).** Shadowed Type-2 Fuzzy Sets in Dynamic Parameter Adaption in Cuckoo Search and Flower Pollination Algorithms for Optimal Design of Fuzzy Fault-Tolerant Controllers. Mathematical and Computational Applications, 27(6), 89. https://doi.org/10.3390/mca27060089
7. **Castillo, O., Amador-Angulo, L. (2018).** A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Information Sciences, 460–461, 476–496. https://doi.org/10.1016/j.ins.2017.10.032
8. **Aguiar, R., Franco, I., Leonardi, F. (2023).** Comparative Analysis of Type-1 and Type-2 Fuzzy Controllers: Exploiting Synergies for Improved Control System Performance. The Journal of Engineering and Exact Sciences, 9(4), 15936–01e. https://doi.org/10.18540/jcecvl9iss4pp15936-01e

9. **Zhang, J., Lin, S., Wang, Y. (2024).** Reformulation and Enhancement of Distributed Robust Optimization Framework Incorporating Decision-Adaptive Uncertainty Sets. Axioms, 13(10), 699. https://doi.org/10.3390/axioms13100699
10. **Huang, C., Yüksel, S., Dincer, H. (2024).** A Novel Fuzzy Model for Knowledge-Driven Process Optimization in Renewable Energy Projects. Journal of the Knowledge Economy. https://doi.org/10.1007/s13132-024-02074-w
11. **D'Aniello, G. (2023).** Fuzzy logic for situation awareness: a systematic review. Journal of Ambient Intelligence and Humanized Computing, 14(4), 4419–4438. https://doi.org/10.1007/s12652-023-04560-6
12. **Sennan, S., Ramasubbareddy, S., Balasubramaniyam, S., Nayyar, A., Abouhawwash, M., Hikal, N. (2021).** T2FL-PSO: Type-2 Fuzzy Logic-Based Particle Swarm Optimization Algorithm Used to Maximize the Lifetime of Internet of Things. IEEE Access, 9, 63966–63979. https://doi.org/10.1109/ACCESS.2021.3069455
13. **Warnakulasooriya, K., Segev, A. (2025).** Comparative analysis of accuracy and computational complexity across 21 swarm intelligence algorithms. Evolutionary Intelligence, 18(1). https://doi.org/10.1007/s12065-024-00997-6
14. **Zhou, X., Ma, H., Gu, J., Chen, H., Deng, W. (2022).** Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. Engineering Applications of Artificial Intelligence, 114. https://doi.org/10.1016/j.engappai.2022.105139
15. **Mohd, N., Besar, R., Aziz, N. (2022).**

    Hybrid Feature Selection of Breast Cancer Gene Expression Microarray Data Based on Metaheuristic Methods: A Comprehensive Review. Symmetry, 14(10), 1955. https://doi.org/10.3390/sym14101955
16. **Sohouli, A., Molhem, H., Zare-Dehnavi, N. (2022).** Hybrid PSO-GA Algorithm for Estimation of Magnetic Anomaly Parameters Due to Simple Geometric Structures. Pure and Applied Geophysics, 179(6–7), 2231–2254. https://doi.org/10.1007/s00024-022-03048-2
17. **Kazerani, R. (2024).** Improving Breast Cancer Diagnosis Accuracy by Particle Swarm Optimization Feature Selection. International Journal of Computational Intelligence Systems, 17(1). https://doi.org/10.1007/s44196-024-00428-5
18. **Bogar, M., Shirodkar, I., Kulkarni, O., Jawade, S., Kakandikar, G. (2024).** Mayfly optimization algorithm: a review. Journal of Mechatronics and Artificial Intelligence in Engineering, 5(1), 17–30. https://doi.org/10.21595/jmai.2024.23909
19. **Owoola, E., Xia, K., Wang, T., Umar, A., Akindele, R. (2021).** Pattern Synthesis of Uniform and Sparse Linear Antenna Array Using Mayfly Algorithm. IEEE Access, 9, 77954–77975. https://doi.org/10.1109/ACCESS.2021.3083487
20. **Patel, H., Sha, V. (2022).** Fuzzy Logic Based Metaheuristic Algorithm for Optimization of Type-1 Fuzzy Controller: Fault-Tolerant Control for Nonlinear System with Actuator Fault. IFAC-PapersOnLine, 55(16), 715–721. https://doi.org/10.1016/j.ifacol.2022.04.117
21. **Zhang, T., Zhou, Y., Zhou, G., Deng, W., Luo, Q. (2022).** Bioinspired Bare Bones Mayfly Algorithm for Large-Scale Spherical Minimum Spanning Tree. Frontiers in Bioengineering and Biotechnology, 10, 830037. https://doi.org/10.3389/fbioe.2022.830037
22. **Lizarraga, E., Valdez, F., Melin, P., Castillo, O. (2025).** A Hybrid Enhanced Mayfly Optimization Algorithm with Improved Performance through Fuzzy-Based Automatic Parameter Adaptation. Computacion y Sistemas, 29(2), 615–631. https://doi.org/10.13053/CyS-29-2-5709
23. **Nagarajan, K., Nanda, K., Rajagopalan, A., Kumar, N., Bajaj, M. (2024).** Improved Mayfly algorithm for optimizing power flow with integrated solar and wind energy," International Journal of Electrical and Electronics Research (IJEER), vol. 12, no. 2, pp. 415–420. https://doi.org/10.37391/IJEER.120212

24. **Roy, S., Jana D., Mishra, A. (2023).** Linguistic interval type-2 fuzzy logic-based exigency vehicle routing: IoT system development for smart city applications with soft computing-based optimization. Franklin Open, 6, 100057. https://doi.org/10.1016/j.fraope.2023.100057
25. **Suzuki, A., Negishi, E. (2024).** Fuzzy logic systems for healthcare applications. Journal of Biomedical and Sustainable Healthcare Applications, 1–9. https://doi.org/10.53759/0088/jbsha20240401
26. **Nishanth, F. P., Dash, S. K., Mahapatro, S. R. (2024).** Critical study of type-2 fuzzy logic control from theory to applications: A state-of-the-art comprehensive survey. E-Prime - Advances in Electrical Engineering, Electronics and Energy, 10. https://doi.org/10.1016/j.prime.2024.100771
27. **Mahmoud, A., Yuan, X., Yua, Y. (2021).** Hybrid Meta-heuristic Adaptive Fuzzy Inference Systems in Rockfill Dam Multi-objective Shape Optimization. KSCE Journal of Civil Engineering, 25(12), 4913–4930. https://doi.org/10.1007/s12205-021-1504-9
28. **Dumitrescu, C., Ciotirnae, P., Vizitiu, C. (2021).** Fuzzy logic for intelligent control system using soft computing applications. Sensors, 21(8). https://doi.org/10.3390/s21082617
29. **Serrano-Guerrero, J., Romero, F. P., Olivas, J. A. (2021).** Fuzzy logic applied to opinion mining: A review. Knowledge-Based Systems, 222. https://doi.org/10.1016/j.knosys.2021.107018
30. **Mendel, J. M. (2014).** General type-2 fuzzy logic systems made simple: A tutorial. IEEE Transactions on Fuzzy Systems, 22(5), 1162–1182. https://doi.org/10.1109/TFUZZ.2013.228641
31. **Yan, S.-R., Alattas, K. A., Bakouri, M., Alanazi, A. K., Mohammadzadeh, A., Mobayen, S., Zhilenkov, A., Guo, W. (2022).** Generalized Type-2 Fuzzy Control for Type-I Diabetes: Analytical Robust System. Mathematics, 10(5). https://doi.org/10.3390/math10050690
32. **Niewiadomski, A., Kacprowicz, M. (2021).** Type-2 fuzzy logic systems in applications: Managing data in selective catalytic reduction for air pollution prevention. Journal of Artificial Intelligence and Soft Computing Research, 11(2), 85–97. https://doi.org/10.2478/jaiscr-2021-0006
33. **Woźniak, M., Szczotka, J., Sikora, A., Zielonka, A. (2024).** Fuzzy logic type-2 intelligent moisture control system. Expert Systems with Applications, 238. https://doi.org/10.1016/j.eswa.2023.121581
34. **Gad, A.G. (2022).** Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review, Archives of Computational Methods in Engineering, 29(5), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4
35. **Jain, V., Saihjpal, N., Singh, N., Singh, S. B. (2022).** An Overview of Variants and Advancements of PSO Algorithm, Applied Sciences, vol. 12, no. 17, article 8392, Aug. https://doi.org/10.3390/app12178392
36. **Chen, Y., Li, C., Yang, J. (2023).** Design and application of Nagar-Bardini structure-based interval type-2 fuzzy logic systems optimized with the combination of backpropagation algorithms and recursive least square algorithms, Expert Syst Appl, vol. 211, Jan. https://doi.org/10.1016/j.eswa.2022.118596
37. **Magaji, N., Bin Mustafa, M.W., Lawan, A. U., Tukur, A., Abdullahi, I., Marwan, M. (2022).** Application of Type 2 Fuzzy for Maximum Power Point Tracker for Photovoltaic System, Processes, vol. 10, no. 8, Aug. https://doi.org/10.3390/pr10081530
38. **Khairuddin, S., Hasan, M., Akhir, E., Hashmani, M. (2022).** "Generating type 2 trapezoidal fuzzy membership function using genetic tuning," Computers, Materials and Continua, vol. 71, no. 1, pp. 717–734. https://doi.org/10.32604/cmc.2022.020666
39. **Zhao,J., Liu, Y., Wang, L., Wang, W. (2020).** A Generalized Heterogeneous Type-2 Fuzzy Classifier and Its Industrial Application, IEEE Transactions on Fuzzy Systems, vol. 28, no. 10, pp. 2287–2301, Oct. https://doi.org/10.1109/TFUZZ.2019.2930492
40. **Bai, Y., Wang, D. (2018).** On the Comparison of Type 1 and Interval Type 2 Fuzzy Logic Controllers Used in a Laser Tracking System,

Elsevier B.V., Jan., pp. 1548–1553. https://doi.org/10.1016/j.ifacol.2018.08.276

41. **Jahanshahi, H., Yousefpour, A., Soradi-Zeid, S., Castillo, O (2022).** A review on design and implementation of type-2 fuzzy controllers, in Mathematical Methods in the Applied Sciences, John Wiley and Sons Ltd. https://doi.org/10.1002/mma.8492
42. **Bhattacharyya, T., Chatterjee, B., Singh, P., Yoon, Z., Geem, W., Sarkar, R. (2020).** Mayfly in Harmony: A new hybrid meta-heuristic feature selection algorithm, IEEE Access, vol. 8, pp. 195929–195945. https://doi.org/ 10.1109/ ACCESS.2020.3031718
43. **Sowan, B., Eshtay, M., Dahal, K., Qattous, H., Zhang, L. (2023).** Hybrid PSO feature selection-based association classification approach for breast cancer detection, Neural Comput Appl, vol. 35, no. 7, pp. 5291–5317, Mar. https://doi.org/10.1007/s00521-022-07950-7
44. **Gabor, T., Phan, T. (2021)**. Linnhoff-Popien, C. Productive Fitness in Diversity-Aware Evolutionary Algorithms. Nat. Comput. 20, 779–795. https://doi.org/10.1007/s11047-021-09853-3
45. **Liu, X., Xu, Y., Gao, W., Xu, Y. (2023).** A Diversity-Guided Ant Colony Algorithm for Permutation Flow Shop Scheduling. Proc. Genet. Evol. Comput. Conf. Companion GECCO '23 Companion, 374–377. https://doi.org/10.1145/3688671.3688759
46. **Puška, A., Nedeljković, M., Zolfani, S., Pamučar, D. (2021).** Application of interval fuzzy logic in selecting a sustainable supplier on the example of agricultural production, Symmetry (Basel), vol. 13, no. 5, May. https://doi.org/10.3390/sym13050774
47. **Gao, Z., Zhao, J., Li, S., Hu, Y. (2020).** The improved mayfly optimization algorithm, in Journal of Physics: Conference Series, IOP Publishing Ltd, https://doi.org/ 10.1088/1742-6596/1684/1/012077
48. **Xu, H., Zhao, M., Xue, F., Zhang, X., Sun, L. (2023).** An Improved Mayfly Algorithm with Shading Detection for MPPT of Photovoltaic Systems, IEEE Access, vol. 11, pp. 110827–110836.https://doi.org/10.1109/ACCESS.2023.3318129
49. **Du, Q., Zhu, H. (2022).** Dynamic elite strategy mayfly algorithm, PLoS One, vol. 17, no. 8, Aug.https://doi.org/10.1371/journal.pone.0273155
50. **Lei, G., Chang, X., Tianhang, Y., Tuerxun, W. (2022).** "An Improved Mayfly Optimization Algorithm Based on Median Position and Its Application in the Optimization of PID Parameters of Hydro-Turbine Governor," IEEE Access, vol. 10, pp. 36335–36349. https://doi.org/10.1109/ACCESS.2022.3160714
51. **Prasanth, V., Ramachandran, M., Ramu, K. "A (2022).** Study on Mayfly Algorithm and Its Recent Developments," Data Analytics and Artificial Intelligence, vol. 2, no. 2, pp. 109–116, Aug. https://doi.org/10.46632/daai/2/2/6
52. **Sedigh, A., Akbarzadeh-T, M.-R., Tomlinson, R. E. (2021).** Comparison of Type-1 and Type-2 Fuzzy Systems for Mineralization of Bioprinted Bone. https://doi.org/10.1101/2021.03.31.437908
53. **Yuan, K., Li, W., Xu, W., Zhan, T., Zhang, L., Liu, S. (2021).** A comparative experimental evaluation on performance of type-1 and interval type-2 Takagi-Sugeno fuzzy models. International Journal of Machine Learning and Cybernetics, 12(7), 2135–2150. https://doi.org/10.1007/s13042-021-01298-5.
54. **Chen, Y. (2022).** Design of sampling-based noniterative algorithms for centroid type-reduction of general type-2 fuzzy logic systems. Complex and Intelligent Systems, 8(5), 4385–4402. https://doi.org/10.1007/s40747-022-00789-4.
55. **Bokharaeian, B., Dehghani, M., Diaz, A. (2023).** Automatic extraction of ranked SNP-phenotype associations from text using a BERT-LSTM-based method. BMC Bioinformatics, 24(1). https://doi.org/10.1186/s12859-023-05236-w.
56. **Kaltak, M., de Bruijn, P., van Leeuwen, W., Platenburg, G., Cremers, F. P. M., Collin, R. W. J., Swildens, J. (2024).** QR-1011 restores defective ABCA4 splicing caused by multiple severe ABCA4 variants underlying Stargardt

disease. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-51203-7

57. **Yin, D., Wu, Z., Yokota, K., Matsumoto, K., Shibayama, S. (2023).** Identify novel elements of knowledge with word embedding. PLoS ONE, 18(6). https://doi.org/10.1371/journal.pone.0284567
58. **Guajardo, H. M., Valdez, F., Melin, P., Castillo, O., Cortes-Antonio, P. (2025).** Comparative Study of Dragonfly and Firefly Algorithms with Type-1 and Type-2 Fuzzy Parameter Adaptation. Advances in Computational Intelligence. MICAI 2024 International Workshops. Lecture Notes in Computer Science, 15464, 3–15. https://doi.org/10.1007/978-3-031-83879-8_1
59. **Carreón-Ortiz, H., Valdez, F., Castillo, O. (2022).** Fuzzy Flower Pollination Algorithm (FFPA): Comparative Study of Type-1 (T1FLS) and Interval Type-2 Fuzzy Logic System (IT2FLS) in Optimization Parameter Adaptation**.** Computación y Sistemas, 26(2), 643–661. https://doi.org/10.13053/CyS-26-2-4247
60. **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 38 (3), 3229-3252. https://doi.org/10.3233/JIFS-191198

Article received on 31/1/2025; accepted on 15/12/2025.
**Corresponding author is Fevrier Valdez.*