www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026, 275-282. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2026.v17i2.872

Hybrid Combination of the Dragonfly and Firefly Algorithms with Parameter Adaptation using Type-2 Fuzzy Logic and its Comparison with Cuckoo Search

Hector M. Guajardo, Fevrier Valdez, Oscar Castillo, Patricia Melin and Prometeo Cortes-Antonio
Tijuana Institute of Technology TecNM
Research and Postgraduate Studies Department,
Engineering Faculty, Instituto Tecnologico de Tijuana,
Tijuana, Baja California, 22414, Mexico.

E-mails: hectormg89@gmail.com, fevrier@tectijuana.mx, ocastillo@tectijuana.mx, pmelin@tectijuana.mx, prometeo.cortes@tectijuana.mx

Abstract. This paper explores a hybrid approach combining the Dragonfly Algorithm (DA) and Firefly Algorithm (FA) to balance exploration and exploitation, avoiding local optima and refining solutions in promising areas. The hybrid achieved higher-quality solutions, faster convergence to optimal results, and adaptability to diverse optimization problems through complementary strategies. Additionally, the Cuckoo Search Algorithm (CS), known for its effectiveness in global optimization via random search and solution space exploitation, was integrated. To further enhance performance, Type-2 Fuzzy Logic was applied for parameter adaptation in the algorithms.
Keywords: Hybrid Algorithm, Cuckoo Search, Type-2 Fuzzy Logic.

Article Info
Received Dec 12, 2024
Accepted Dec 12, 2025

1 Introduction

Bio-inspired optimization methods, such as the Dragonfly Algorithm (DA) (Wikelski et al., 2006; Mirjalili, 2016), Firefly Algorithm (FA) (Yang, 2010), and Cuckoo the Search (CS) Algorithm (Yang, 2014), have proven highly effective for tackling complex optimization problems. These algorithms, inspired by natural behaviors, perform exceptionally well in scenarios where exhaustive search methods are impractical and the likelihood of being trapped in local optima is significant. Their strength lies in combining in an appropriate fashion global exploration with local exploitation, allowing adaptability to dynamic and challenging problems. The Dragonfly Algorithm leverages the movement patterns of dragonflies to maintain a balance between exploration and exploitation, making it particularly suitable for both continuous and discrete optimization tasks while preserving diversity and avoiding local optima. In contrast, the Firefly Algorithm, inspired by bioluminescent attraction, enhances convergence toward global optima by selectively favoring solutions based on intensity, making it ideal for multi-peak and nonlinear problems. By integrating the strengths of DA and FA, a hybrid approach is proposed that balances exploration, adaptability, and convergence efficiency, making it robust for high-dimensional optimization challenges. Additionally, algorithms like Differential Evolution (DE) (Storn & Price, 1997), Particle Swarm Optimization (PSO) (Sengupta et al., 2018), and the Genetic Algorithm (GA) (Saophan et al., 2023), are recognized for their efficiency in speed and convergence. Finally, we got the cuckoo search algorithm, which is inspired by the nesting behavior of cuckoos, which lay their eggs in the nests of other birds. The algorithm combines random search with intensive exploitation of promising areas in the solution space. Tacking some examples of hybrid approaches (Barraza et al., 2018) those are robust and versatile strategies, ideal for addressing complex, high-dimensional problems where flexibility and precision are crucial. These strategies, when combined, leverage the individual algorithm strengths to improve solution quality and convergence speed. As studies suggest, systems modeled on natural behaviors of insects, such as dragonflies, fireflies, and damselflies, exhibit remarkable functional efficiency. Metaheuristic techniques skillfully integrate various strategies to navigate complex solution landscapes, optimizing objectives such as efficiency maximization or cost minimization. The incorporation of probabilistic and deterministic principles allows DA, FA, and DE to address real-world optimization problems effectively.

This article is organized as follows: Section 2 discusses nature-inspired optimization techniques; Section 3 reviews relevant literature about Fuzzy Logic (Zadeh, 1965; Kosko, 1992); Section 4 overviews Type-2 Fuzzy Logic (Carreon-Ortiz et al., 2023), Section 5 details the results and Section 6 provides the study's conclusions and summary.

2 Nature Inspiration

The inspiration for these algorithms began when some researchers started observing nature and that is how the Dragonfly, Firefly and Cuckoo Search algorithms were developed, and each of these algorithms has certain defining characteristics. Starting with the Dragonfly Algorithm, this one in particular has the following characteristics: separation, avoiding collisions between individuals in the swarm, alignment: maintaining a uniform direction with neighboring individuals, cohesion: attracting dragonflies to the center of the swarm, attraction to food sources: guiding the swarm toward desirable targets, and enemy repulsion: moving away from potential threats. This one presents the next advantages: Efficient balance between exploration and exploitation, a high adaptability to both continuous and discrete problems, and lastly maintains diversity of solutions to avoid local optima.

The Firefly Algorithm is inspired by the bioluminescent attraction behavior of fireflies. The algorithm leverages the concept that brighter fireflies attract others, simulating an optimization process guided by intensity. Fireflies are unisex and can attract others regardless of gender, the attraction between fireflies is proportional to their brightness; less bright fireflies move toward brighter ones, and the brightness of a firefly is determined by the quality of the solution it represents. This algorithm presents the next advantages a rapid convergence to global optima, effective in solving nonlinear and multi-modal problems, and simple implementation with few adjustable parameters.

The last one is the Cuckoo Search Algorithm based on the nesting behavior of cuckoos, which lay their eggs in the nests of other birds. The algorithm combines random search with intensive exploitation of promising areas in the solution space. This one counts with the advantage of being highly efficient in global search tasks, it's capable of escaping local optima through long exploratory steps, and minimal parameters, simplifying implementation.

2.1 Dragonfly Algorithm

The Dragonfly Algorithm presents the following mathematical representation.

$$S_j = \sum_{j=1}^{N} X = X_j \tag{1}$$

Alignment: Individual velocity matched with neighbor individuals.

$$A_j = \sum_{J=1}^{N} V_j \tag{2}$$

Cohesion: Individual tendency toward center of the herd.

$$C_j = \frac{\sum_{j=1}^{N} X_j}{N} \text{ - X} \tag{3}$$

Attraction to food source is calculated.

$$F_j = X^+ - X \tag{4}$$

Where:

X Is the position of the current individual.
X^+ Is the area of the food.

Distraction from enemy is calculated:

$$F_j = X^- + X \tag{5}$$

Where:

X^+ Is the position of the current individual.

X^- Is the area of the enemy.

2.2 Firefly Algorithm

The firefly algorithm presents the following mathematical representations

$$X_p^t + 1 = X_p^t + \beta(r)\left(X_p - X_q\right) + (rand - \tfrac{1}{2}) \tag{6}$$

2.3 Hybrid DA-FA Algorithm

The Hybrid algorithm created with the Dragonfly and Firefly was constructed by combining the best characteristics of each of these two algorithms. For the Dragonfly we select exploration, which provides the ability to thoroughly search the solution space, avoiding local optima by balancing multiple dynamic forces. For the Firefly Algorithm we select the exploitation, which ensures a focused search around promising regions, improving convergence to high-quality solutions. Together, these principles can form a hybrid algorithm that leverages the strengths of both exploration and exploitation. The process begins by setting up the key parameters: alpha and gamma for the Firefly algorithm, which help control how fireflies move and attract each other, and W, C1, and C2 for the Dragonfly algorithm, which manage the balance between exploring new areas and focusing on promising ones. To make the algorithm more flexible and precise, Type 1 and Type 2 fuzzy logic systems were added. These systems allow the algorithm to adjust dynamically based on the situation, making it more responsive to changes. The real strength of DAFA lies in how it combines the best parts of the two algorithms. It uses the exploration phase from the Dragonfly algorithm to search broadly by updating the positions and velocities of the population. Then, it switches to the exploitation phase of the Firefly algorithm to fine-tune the search and zero in on the best possible solution. By blending these approaches, DAFA successfully finds the optimal solution, showing how effective this combination can be. The result is a balanced and powerful algorithm designed to handle even the most challenging optimization problems with ease. We can observe the algorithm in Figure 1.

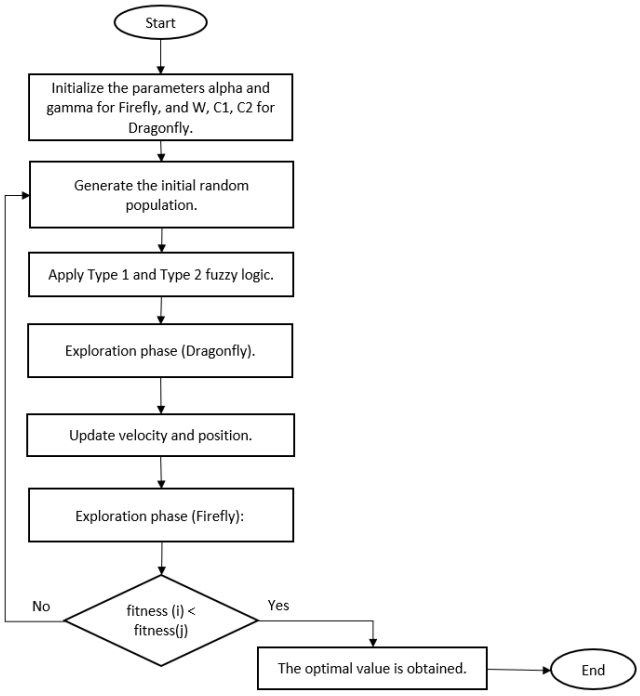

Fig. 1. Represents the Hybrid DAFA Algorithm.

2.4 Cuckoo Search

The Cuckoo Search (CS) (Yang, 2014) Algorithm presents the next mathematical representations for nesting parasitism. The principle of nesting parasitism is modeled by generating new solutions $x_i^{(t+1)}$ and replacing the worst-performing ones based on fitness. This can be expressed as:

$$X_i^{(t+1)} = X_t^{(x)} + \alpha \,.\, (X_{best} - X_i^{(x)}) \tag{7}$$

Where:

$x_i^{(t)}$: The current solution in the $i-th$ psoition at iteration t.
x_{best} : The best solution found so far.
α:A step-size scaling parameter(controls the learning rate).
If $x_i^{(t+1)}$ outperforms the current worst solution, it replaces it.

The worst-performing solutions are determined by their fitness values, and the algorithm keeps the best-performing solution to guide the search process.

$$x_I^{(t+1)} = x_I^t + \alpha \,.\, L(s) \tag{8}$$

Where:

$x_I^{(t+1)}$: the updated solution for the $i-th$ position at iteration $t+1$.
α: A scaling parameter controlling the step size.
$L(s)$: A levy floght step, defined as:

$$L(s) \sim \frac{1}{|s|^{\lambda'}} \; 1 < \lambda \leq 3 \tag{9}$$

To approximate Levy flights in computational implementations, $L(s)$ can be generated using Mantegna's algorithm:

$$L(s) = \frac{u}{|u|^{\frac{1}{\lambda}}} \tag{10}$$

u and v are drawn from normal distributions:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_u^2) \tag{11}$$

These steps enable CS to effectively explore the search space while maintaining diversity and avoiding local optima.

3 Fuzzy Logic

As we know, fuzzy logic (Zadeh, 1965; Kosko, 1992), mimics human reasoning to handle uncertainty and imprecision. Unlike classical logic, which operates with binary values (true or false, 0 or 1), with fuzzy logic we have the particularity of applying degrees of truth or memberships, ranging from 0 to 1 (Klir & Yuan, 1995). Using this theory, we have applied Type-1 fuzzy logic to the DA-FA algorithm for adapting 2 and beta parameters with the following characteristics: Input as "iteration" and the output as w or beta depending the case. We select 3 membership functions in the input and 3 membership functions in the output, and we apply three fuzzy rules: if iteration is low, then 2 and beta will be high, if iteration is medium, then w and beta will be medium, and if iteration is high, then w and beta will be low. Figure 2 depicts the fuzzy system for the w parameter specifying the interaction as input with the 3 membership functions, the w as output with 3 membership functions and the fuzzy rules that explain the final results. Figures 3 exhibits the Beta parameter that explain the input with 3 membership functions the output with the same amount of membership functions and the fuzzy rules applied to the Beta parameters, Figure 4 illustrates the membership functions in which the colors indicate the fuzzy rule if the color is blue the rule will be low, if the color is red the rule will be medium and finally if the color is yellow the rule will be high.

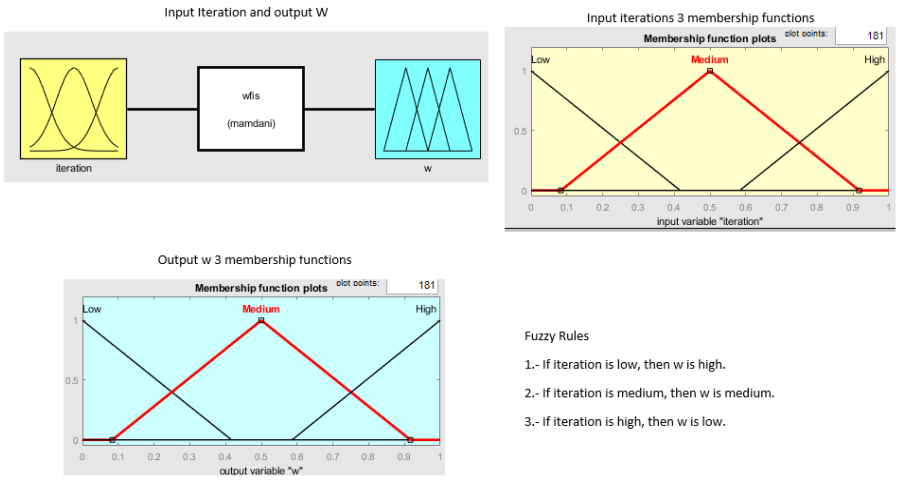

Fig. 2. Represents the specifications of the W Parameter.

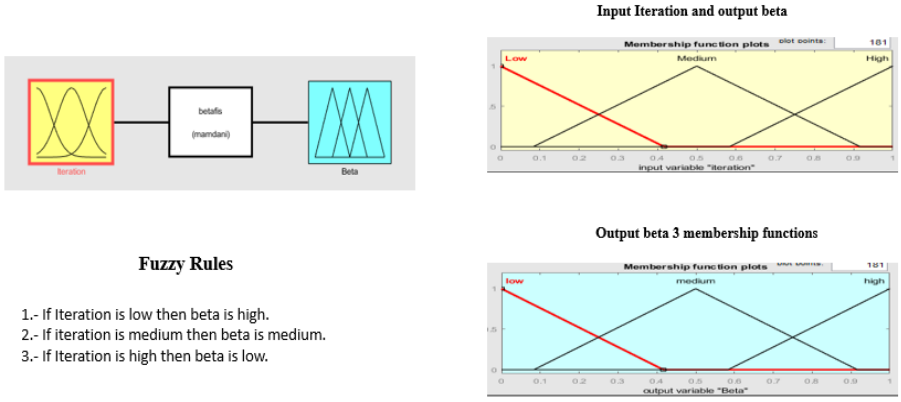

Fig. 3. Represent the specifications of Beta parameter.

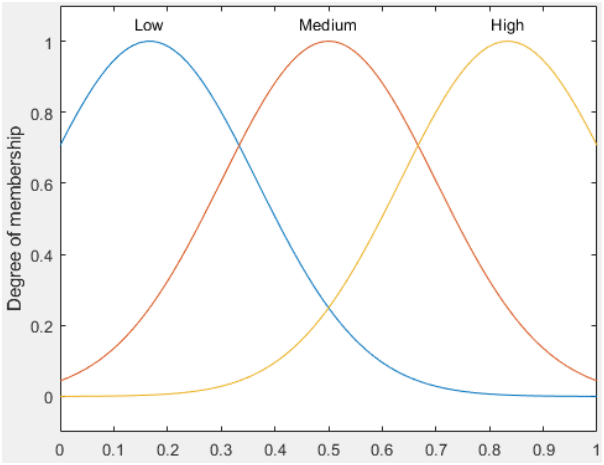

Fig. 4. Type-1 Membership functions.

4 Fuzzy Logic

Type-2 fuzzy logic (T2FL) (Yang, 2014), is an advanced form of fuzzy logic designed to handle higher levels of uncertainty and imprecision. While traditional Type-1 fuzzy logic uses precise membership functions, Type-2 fuzzy logic extends this by incorporating a degree of uncertainty into the membership functions themselves. Applying the aforementioned framework, we

applied Type-2 Fuzzy Logic to the DA-FA and Cuckoo Search algorithms, with the next characteristics: Gaussian input as "iteration" and output as w or beta depending the case. We select 3 membership functions in the input, and 3 membership functions in the output and we apply three fuzzy rules: if iteration is low, then w and beta will be high, if iteration is medium, then w and beta will be medium, and if iteration is high, then w and beta will be low.

4.1 Mamdani Centroid Defuzzifier

The centroid defuzzifier combines the type-1 fired-rule output fuzzy sets using union, and then finds the centroid, $y_c(x')$.

$$y_c(x) = \frac{\sum_{i=1}^{N} y_i \mu_\beta(y_i|x^1)}{\sum_{i=1}^{N} \mu_\beta(y_i|x^1)} \tag{12}$$

4.2 Type-2 Fuzzy Sets

A type-2 fuzzy set can be postulated as follows:

$$\tilde{A} = \{\big((x,u), \mu_{\tilde{A}}(x,u) | x \in X, u \in\equiv [0,1]\big)\} \tag{13}$$

A type-2 fuzzy sets (Mendel, 2017) (also called general type-2 fuzzy set), denoted ~A, is the graph of a bivariate function called the MF of ~A on the Cartesian product X [0,1] into [0,1], where X is the universe for the primary variable of ~A, x. The MF of ~A is denoted $\mu_{\tilde{A}}(x,u)$, or $\mu_{\tilde{A}}$ for short, and is called a type-2, the red color indicates the upper limit meantime the blue color indicates the lower limit and the gray color indicates the footprint of uncertainty, we can review this in Figure 5.

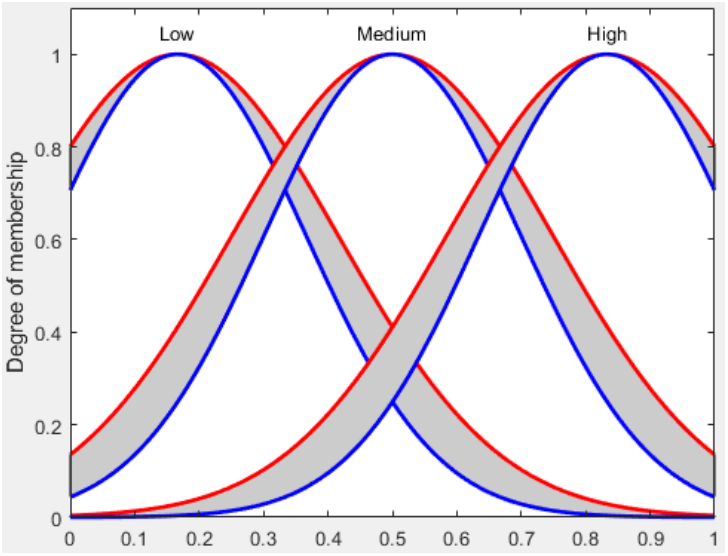

Fig. 5. Type-2 membership functions.

5 Results

The experiments were performed under uniform conditions and applied to the first 5 functions in the case of DA-FA, and the original CS, both with type 2 fuzzy logic. The DA-FA algorithm demonstrated the following specifications when applied to the five mathematical functions: a population size of 40 agents, a maximum of 500 iterations, and a problem dimensionality of 1000. Similarly, the CS algorithm exhibited these characteristics: a population size of 40 agents, a maximum of 500 iterations, and a problem dimensionality of 100. A comparative analysis between DA-FA and the CS algorithm, both incorporating Type-2 Fuzzy Logic, is summarized in Table 1.

Table 1. Results of comparison from DAFA with Type-2 and Cuckoo Search with Type-2 with 100 dimensions.

			DAFA			CS
	Type-2 F1	Type-2 F2	Type-2 F3	Type-2 F4	Type-2 F5	Type-2 OG
	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40
	Iterations:500	Iterations:500	Iterations:500	Iterations:500	Iterations:500	Iterations:500
	1000 Dim	1000 Dim	1000 Dim	1000 Dim	1000 Dim	100 Dim
AVG	2.64E+06	2.63E+147	9.66E+199	7.27E+01	5.30E+08	8.69E-06
STD	1.10E+06	8.29E+147		3.69E+00	3.16E+08	1.41E-06

The experiments were performed under the same conditions for consistency. The DAFA algorithm, was evaluated using five mathematical functions, with a setup consisting in 40 agents in the population, a limit of 500 iterations, and a problem dimensionality set to 1000. In parallel, the CS algorithm was tested under similar parameters: a population size of 40 agent, a maximum of 500 iterations, and a dimensionality of 1000. Table 2 highlights the comparison between the DAFA and the CS, both of which incorporate Type-2 Fuzzy Logic.

Table 2. Results of comparison from DAFA with Type-2 and Cuckoo Search with Type-2 with 1000 Dimensions.

			DAFA			CS
	Type-2 F1	Type-2 F2	Type-2 F3	Type-2 F4	Type-2 F5	Type-2 OG
	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40
	iterations:500	iterations:500	iterations:500	iterations:500	iterations:500	iterations:500
	1000 Dim	1000 Dim	1000 Dim	1000 Dim	1000 Dim	1000 Dim
AVG	2.64E+06	2.63E+147	9.66E+199	7.27E+01	5.30E+08	3.30E-12
STD	1.10E+06	8.29E+147		3.69E+00	3.16E+08	2.96E-12

The experiments were executed under consistent conditions, regarding the DAFA algorithm, it has the following characteristics when applied to the 5 mathematical functions: involving a population of 40 agents, a maximum of 500 iterations, and a problem dimensionality of 1000. Regarding the CS algorithm, it has the following characteristic: involving a population of 40 agents, a maximum of 500 iterations, and a problem dimensionality of 2000. Table 3 presents the comparisons between DAFA and CS, both with Type-2 Fuzzy Logic.

Table 3. Results of comparison from DAFA with Type-2 and Cuckoo Search with Type-2 with 2000 Dimensions.

			DAFA			CS
	Type-2 F1	Type-2 F2	Type-2 F3	Type-2 F4	Type-2 F5	Type-2 OG
	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40	Agents: 40
	iterations:500	iterations:500	iterations:500	iterations:500	iterations:500	iterations:500
	1000 Dim	1000 Dim	1000 Dim	1000 Dim	1000 Dim	2000 Dim
AVG	2.64E+06	2.63E+147	9.66E+199	7.27E+01	5.30E+08	2.63E-25
STD	1.10E+06	8.29E+147		3.69E+00	3.16E+08	2.94E-25

6 Conclusions

After designing and implementing the hybrid algorithm, the next step involved integrating Y2FL for adaptation of parameters. This enhancement aimed to improve the algorithm's performance by addressing uncertainties in the parameter space. Subsequently, a comprehensive set of experiments was conducted using benchmark functions labeled F1 through F5. These functions are widely recognized in optimization research for evaluation algorithmic efficiency and effectiveness. The data collected from these experiments provided insights into the algorithm's behavior across various optimization scenarios.

To assess the effectiveness of the proposed approach, the performance of the hybrid algorithm with T2FL was systematically compared to that of the original CS algorithm. This comparison specifically focused on the impact of parameter adaptation using T2FL in achieving better convergence and solution quality, highlighting the advantages of the enhanced method over its baseline counterpart. As a preliminary conclusion, based on the experiments conducted so far using benchmark functions F1 through F5, the CS algorithm with the application of T2FL demonstrates superior performance when compared to the hybrid DA-FA algorithm also enhanced with T2FL. The results indicate that the CS algorithm achieves better outcomes, reinforcing its potential effectiveness and reliability in addressing optimization challenges.

References

Barraza, J., Rodríguez, L., Castillo, O., Melin, P., & Valdez, F. (2018). A new hybridization approach between the Fireworks Algorithm and Grey Wolf Optimizer Algorithm. *Mathematical Problems in Engineering, 2018*, 6495362. https://doi.org/10.1155/2018/6495362

Carreon-Ortiz, H., Valdez, F., & Castillo, O. (2023). Comparative study of type-1 and interval type-2 fuzzy logic systems in parameter adaptation for the fuzzy discrete mycorrhiza optimization algorithm. *Mathematics, 11*(11), 2501. https://doi.org/10.3390/math11112501

Castillo, O. (2012). Type-2 fuzzy logic. En O. Castillo, *Type-2 fuzzy logic: Theory and applications* (pp. 21–35). Springer. https://doi.org/10.1007/978-3-642-24663-0_2

Khennak, I., Drias, H., Drias, Y., et al. (2023). I/F-Race tuned firefly algorithm and particle swarm optimization for K-medoids-based clustering. *Evolutionary Intelligence, 16*, 351–373. https://doi.org/10.1007/s12065-022-00794-z

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall.

Kosko, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. Prentice-Hall.

Mendel, J. M. (2017). *Uncertain rule-based fuzzy systems: Introduction and new directions* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-51379-7

Saophan, P., Pannakkong, W., Singhaphandu, R., & Huynh, V. (2023). Rapid production re-scheduling for flow shop under machine failure disturbance using hybrid perturbation population genetic algorithm-artificial neural networks (PPGAANNs). *IEEE Access, 11*, 75794–75817. https://doi.org/10.1109/ACCESS.2023.3294573

Sengupta, S., Basak, S., & Peters, R. A. (2018). *Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives*. *Machine Learning and Knowledge Extraction, 1*(1), 157–191. https://doi.org/10.3390/make1010010

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(3), 341–359. https://doi.org/10.1023/A:1008202821328

Wikelski, M., Moskowitz, D., Adelman, J. S., Cochran, J., Wilcove, D. S., & May, M. L. (2006). Simple rules guide dragonfly migration. *Biology Letters, 2*(3), 325–329. https://doi.org/10.1098/rsbl.2006.0487

Yager, R. R., & Filev, D. P. (1994). *Essentials of fuzzy modeling and control*. Wiley-Interscience.

Yang, X.-S. (2010). Firefly algorithms for multimodal optimization. En *Stochastic Algorithms: Foundations and Applications* (Lecture Notes in Computer Science, Vol. 5792, pp. 169–178). Springer. https://doi.org/10.1007/978-3-642-04944-6_14

Yang, X.-S. (2014). Cuckoo search and firefly algorithm: Overview and analysis. En *Cuckoo Search and Firefly Algorithm* (pp. 1–26). Springer. https://doi.org/10.1007/978-3-319-02123-0_1

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353.

Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications, 27*(4), 1053–1073. https://doi.org/10.1007/s00521-015-1920-1