Abstract: The objective of this work is focused on improving the optimization ability of the cuckoo search algorithm (CS), and, for this reason, a comparison is made between type-1 and interval type-2 fuzzy logic to look for more promising results in the cuckoo search algorithm (CS), and to help performance, we dynamically adjust the alpha parameter. The idea is to enable CS in leaving the local optima, and then be able to reach the global optima. Currently, there are good results in improving the optimization of algorithms through intelligent fuzzy logic computing after finding the best adjustment parameters. The approach is based on finding the ideal rules with their respective linguistic variables to represent the real world as is perceived by humans. The membership functions that the fuzzy system uses are symmetrically defined for reducing the search space, and this symmetry is what makes the algorithm efficient. We plan to test the proposal in future works in the optimal design of control systems. In the present study, we use five benchmark mathematical functions with variation in the number of dimensions to validate the approach and perform the comparison of interval type-2 and type-1 fuzzy systems in parameter adaptation. For the dynamic adjustment of the parameters, we select the alpha parameter, and the values of Pa and Beta are defined based on the analysis of their behavior in previous works.

Keywords: cuckoo search; interval type-2; type-1 fuzzy system

1. Introduction

In our daily life, there are various tasks that make our activities easier; for this, we can perceive something as basic as using a washing machine, a car, a digital camera, an elevator, etc. that allow us to carry out an activity in less time. In all of these examples, we can denote the implementation of soft computing, which is a branch of Artificial Intelligence (AI) that allows us, through mathematical tools, to implement the way in which we, as human beings, process or manage activities or reasons to recognize objects or people and make decisions. Fuzzy logic is a relatively simple method of information processing, decision making, and process control.

In this article, we will address the concept of fuzzy logic, the contrast between type-1 and type-2 fuzzy, their basic concepts, and the applications that we can find in modern times.

The main motivation of this study is to provide a better performance in CS, and then to be able to implement in future works in different control systems, contribute to the optimization, and compare the results with other metaheuristics.

The idea of fuzzy logic was put forward by Professor L.A. Zadeh of the University of California, Berkeley, in 1965, who published an article called “fuzzy set”, which initiated a new area that is now known worldwide [1,2].

Fuzzy logic has been implemented with bio-inspired algorithms for the optimization of controllers for solving a wide range of problems, such as the case of the galactic swarm algorithm, particle swarm optimization, and hybrid optimization algorithms, in the control...
for rail vehicle systems [3], and control problems [4–9], bio-inspired algorithms have also been compared [10,11] in different applications.

The CS algorithm has been used for capacitated vehicle routing [12], optimal tuning of PID controllers [13], optimal heat exchanger network synthesis [14], imaging systems [15,16], selection of elective subjects in schools [17], in computer vision technology [18], multi-threshold image segmentation [19], and the COVID-19 problem [20], among others.

In a review of the literature, we can find different variants of the CS algorithm with improved beta distribution (IBCS); this parameter is found in the step size in the Lévy flights, which shows a beta distribution in the evolutionary process [21], chaotic cuckoo search [22], the no-linear optimization in CS [23], numerical analysis [24], and Gauss Cauchy with fuzzy logic [25].

The algorithms that implement type-2 fuzzy logic have shown to improve the results [26,27] with respect to the performance of the original versions of the algorithms by increasing the complexity of the dimensions of the problem [28], for example, in control problems [6,13,29,30], which is why we are interested in evaluating a comparison with respect to the original CS algorithm and proving its effectiveness with respect to the proposal.

Different algorithms have implemented type-1 and type-2 fuzzy logic to improve the results with respect to the original algorithm; we know that there are currently several complex algorithms, and that there is no ideal algorithm that fits all real-world problems. This is why different researchers continue to search for new alternatives and proposals for algorithms that implement optimization and thereby improve their performance, reducing time or computational cost; therefore, this is the main objective of our research.

The present work is novel, since the vast majority of investigations, at the moment, only use the CS algorithm, without performing the parameter analysis. The alpha parameter is implicit in the mathematical equation that involves the step size, and if it is not well balanced, it could be bad for the algorithm. However, thanks to the implementation of the type-1 or type-2 fuzzy system, we guide the solutions and obtain promising results.

The present paper includes the following organization. Section 2 presents basic concepts of fuzzy theory and the difference between type-1 and type-2 fuzzy logic, as well as the applications that exist so far. Later, we make an introduction to the concept and operation of the algorithm CS in Section 3. In Section 4, we address the issue of the proposed approach. In Section 5, the benchmark functions are presented, and we describe each of the functions at a mathematical level, as well as the parameters that we consider in the files. In Section 6, we talk about the results, and finally, the conclusions are made in Section 7, where we visualize the results of the statistical tests and foresee future lines of research.

2. Basic Concepts of Fuzzy Logic

To talk about fuzzy sets, it is necessary to describe the concept of the universe of discourse, which is defined by having a range of values which we will call U; for example, U = En, where En is a space of n dimensions, and U is a universe.

The fuzzy set is defined on a universe, which considers taking values in the range of real numbers between zero and one [31,32].

The linguistic variable is a variable that includes a word or sentence in natural language or understanding in artificial terms. For example, to describe a person’s age as a child, young person, or older adult, age will be considered a linguistic variable; instead of using a number, more specific terms such as child, young person, or older adult would be used. In a universe, you have a collection set of linguistic values. Each linguistic value will be associated with a semantic rule; the meaning of a linguistic value is characterized by a function in such a way that a 15-year-old person may have a value of 0.4 that allows him to be compatible with the young linguistic variable. In particular, treating the word young as a linguistic variable with values such as not very young, not very young leads
us to what is called fuzzy logic due to human reasoning that traditional logic could not provide. Therefore, linguistic variables have been found in the field of expert systems in the field of artificial intelligence, as well as pattern recognition, the learning of our tastes in social networks, or in online purchases to offer us products that are attractive to us, as well as medical diagnosis, predictions, the optimization of economic resources, time, natural resources in production lines, among other related areas.

In Figure 1, in the first image, we visualize the classic set theory to know if a person is tall, in a set to which people with a height of 1.8 m would belong. In the second image, we visualize the concept of a fuzzy set for a tall person, in which its approach does not clearly specify by means of a border determining whether or not it belongs to the tall category; said value is in a set of 0 and 1; that is, a person who measures 1.50 could belong to the fuzzy set of tall people with a certain degree.

![Difference between traditional set and type-1 fuzzy sets.](image)

**Figure 1.** Difference between traditional set and type-1 fuzzy sets.

A fuzzy system must have input data, a fuzzifier, a fuzzy inference mechanism through rules (IF-then), a defuzzifier, and must represent a numerical value in the output.

In the literature, we find the introduction to fuzzy systems as a communication mechanism between humans [33], and computing and novel methods for decision making incorporating fuzzy logic that help improve the CPU [34] and the system for climatic decision making to implement in tunnels; this new application uses fuzzy logic and a fuzzy neuro system [32]. Emerging intelligent computing today combines geographic fuzzy logic in a spatial context [35].

Starting from the concept of type-1 fuzzy logic of Zadeh, type-2 fuzzy logic allows us to manage linguistic and numerical uncertainty. Figure 2 shows the difference with respect to type-1 and type-2. The difference is that in type-2, it is required to have a type reducer in the output process to derive a type-1 fuzzy set from a type-2 fuzzy set, such that the gray area represents the footprint of uncertainty, which represents the union of all memberships.

![Difference between fuzzy logic of type-1 and type-2.](image)

**Figure 2.** Difference between fuzzy logic of type-1 and type-2.

By implementing a footprint of uncertainty to the logical function using fuzzy type-2, it is expected that the results will improve significantly. The gray color shows the amplitude of the uncertainty footprint. Visually, it allows us to identify the main difference with respect to type 1 fuzzy logic, which does not have a trace of uncertainty and type 2 fuzzy logic, which does include it.
The main contribution of this work is the proposal of a type-2 fuzzy parameter adaptation in the cuckoo search algorithm, to improve its efficiency, which has not previously been presented in the state of the art. In addition, the application of type-2 fuzzy parameter adaptation in the CS in function optimization is presented.

Currently, there are several investigations that have implemented type-1 fuzzy logic to help diagnose some diseases that help doctors improve early detection diagnoses of various diseases; some cases are: infarction [36], autism [37], and chronic kidney issues [38].

Fuzzy set theory gives way to the set of fuzzy rules from the knowledge of the linguistic variables that allows us to know the degree of membership from the result that we obtain during the process of fuzzification and defuzzification.

3. CS Algorithm

In [14,39], the CS algorithm via Lévy flights was presented. Cuckoos can mimic the behavior of other bird species, such as nestlings. Cuckoos have a peculiar approach to motherhood; they put their eggs in so-called communal nests, which house their eggs, to increase the chance of survival, and cuckoo birds camouflage their eggs in both color and pattern [39–42].

The CS includes randomness to select new solutions, allowing exploration in other search spaces [39].

The CS method has been used in several applications; many researchers continue to propose new variants of the algorithm, but CS needs different parameter values [42].

To achieve good results with the CS algorithm, it is necessary to dynamically control its parameters [39,41]:

1. The Pa is the discovering probability, which helps us to replace nests with no reasonable solutions.
2. The beta parameter serves as a scale factor; this variable is implicit in the Lévy flight equation for exploring and exploiting the search space.
3. The alpha parameter is a factor for convergence speed.

There are other methods similar to the cuckoo search algorithm that, instead of selecting nests, instead have a probability of being selected, and the optimization is based on the evolution based on symmetry; that is, some will have a higher probability than others [43,44].

For generating a new solution, \( x_{(t+1)}^i \), a Lévy flight is given by Equation (1):

\[
x_{(t+1)}^i = x_t^i + \alpha \cdot \text{randn}(\text{size}(D)) \cdot \text{Lvy}(\beta) \cdot s_i
\]

where:
- \( x_{(t+1)}^i \): the new position;
- \( x_t^i \): current position;
- \( \alpha \): scale of the problem;
- \( \text{randn} \): a random;
- \( D \): is the dimensions;
- \( \text{Lvy}(\beta) \): the Lévy flight.

\[
S_i = (x_t^i - x_{\text{best}})
\]

\( x_t^i \): a position in the vector of nest we are evaluating;
\( x_{\text{best}} \): best position.
If \( x_t^i = x_{\text{best}} \), the solution is preserved.

Lévy flight:

\[
\text{Lvy}(\beta) = \frac{\sigma(\beta) \cdot \text{randn}}{|\text{randn}|^\beta}
\]
where:

\[ \beta : \text{a constant (}1 \leq \beta \leq 2). \]

Present standard deviation:

\[
\sigma(\beta) = \begin{cases} 
\Gamma(1 + \beta) \sin \left( \frac{\pi \beta}{2} \right) & \quad \beta \neq 1, 2 \\
\Gamma \left( \frac{3}{2} \right) \beta^{2} \left( \frac{1}{\beta - 1} \right) & \quad \beta = 1, 2
\end{cases}
\]

The gamma function is shown below:

\[
\Gamma(z) = \int_{0}^{\infty} t^{z-1} e^{-t} \, dt \quad (5)
\]

Formulas (1) through (5) are required for the CS algorithm.

CS Algorithm

In this section, we will address the explanation through a flowchart regarding the steps that the CS algorithm follows.

The CS algorithm is a bio-inspired method based on nature; however, we can transfer it to a computational level; in Figure 3, the steps that this algorithm follows are shown.

![Figure 3. Flowchart of the CS algorithm.](image-url)
4. Proposed Approach

This section shows the approach of our work, which shows how our fuzzy systems achieve parameter adaptation.

In Figure 4, the proposed algorithm is presented, where we can appreciate that the CS algorithm will be implemented with type-1 fuzzy logic (FCS1) and interval type-2 fuzzy logic (FCS2). The experiments were performed with a set of benchmark functions.

4.1. Type-1 Fuzzy Logic

This diagram summarizes the operation of type-1 fuzzy inference systems. Figure 5 represents a Mamdani type-1 fuzzy system, which, when applied to the CS algorithm, we called FCS1; it includes one crisp input variable and output variable. The fuzzy input set enters a fuzzy inference system that contains if-then rules. Subsequently, the fuzzy output set enters into a defuzzifier, which has crisp outputs (which results in a single real number).

4.2. Type-2 Fuzzy Logic

This diagram illustrates the operation of the type-2 fuzzy inference system. Figure 6 shows the interval type-2 fuzzy approach for the FCS2, where one input and two outputs are possible, as the final output result is a real number.

A FCS1 FIS comprises a fuzzifier, inference system, rules, and defuzzifier, and FCS2 FIS in its output includes a type reducer [30].
5. Benchmark Functions

In this section, the different benchmark functions that were used to validate the CS algorithms are mathematically described.

In Table 1 show the mathematical functions used in this work.

Table 1. Mathematical functions used in this work.

| Name          | Function                                                                 |
|---------------|--------------------------------------------------------------------------|
| Rastrigin     | \( f(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10 \cos(2\pi x_i)) \)            |
| Griewant      | \( f(x) = \sum_{i=1}^{d} \frac{x_i^2}{400} - \prod_{i=1}^{d} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \) |
| Ackley        | \( f(x) = -20 \exp \left( -0.2 \sqrt{\sum_{i=1}^{d} x_i^2} \right) - \exp \left( \frac{1}{\pi} \sum_{i=1}^{d} \cos(2\pi x_i) \right) + 20 + e \) |
| Sphere        | \( f(x) = \sum_{i=1}^{n} x_i^2 \)                                        |
| Rosenbrock    | \( f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \)    |

To carry out the experiments, we have considered five benchmark mathematical functions with different dimensions to measure the efficiency and performance of the proposed method.

6. Experimental Results

To carry out the set of experiments, it is necessary to specify the parameters that were used, as well as a description of the population and the dimensions used; we address the issue below.

Table 2 shows the used parameters with the experiments for CS, FCS1, and FCS2, with a population of 100 nests for 8, 16, 32, 64, and 128 dimensions.

Table 2. Parameter settings.

| Parameters | CS       | FCS1     | FCS2     |
|------------|----------|----------|----------|
| Pa         | 75%      |          |          |
| Beta       | 1.5      |          |          |
| Alpha      | 5%       | Dynamic  | Dynamic  |

In order to suggest the values of the parameters to be controlled in the fuzzy system, we have previously carried out a study of the parameters [41].
Table 2 shows the experimental results obtained in the simulations, as well as the original CS; FCS1 and FCS2 means are compared, and standard deviations are shown for each function, with 8, 16, 32, 64, and 128 dimensions.

Table 3 shows the results obtained when performing the tests with the sphere benchmark function.

Table 3. Results: the CS, FCS 1, and FCS 2 using the Rosenbrock function.

| Average | Standard Deviation |
|---------|--------------------|
| Dim.    | CS     | FCS1 | FCS2 | Dim.    | CS | FCS1 | FCS2 |
|---------|--------|------|------|--------|----|------|------|
| 8       | 6.62 × 10^{-8} | 9.67 × 10^{-12} | 2.05 × 10^{-11} | 8 | 1.17 × 10^{-7} | 2.40 × 10^{-11} | 4.94 × 10^{-11} |
| 16      | 5.75 × 10^{-1} | 5.68 × 10^{-2} | 7.79 × 10^{-2} | 16 | 6.05 × 10^{-1} | 6.65 × 10^{-2} | 7.55 × 10^{-2} |
| 32      | 1.91 × 10^{1}  | 1.75 × 10{1} | 1.89 × 10{1} | 32 | 4.81 × 10^{0}  | 6.68 × 10^{-1} | 3.45 × 10^{-1} |
| 64      | 6.10 × 10^{1}  | 5.60 × 10{1} | 5.78 × 10{1} | 64 | 3.87 × 10^{1}  | 3.00 × 10^{-1} | 2.21 × 10^{-1} |
| 128     | 3.66 × 10^{2}  | 1.25 × 10^{2} | 1.29 × 10^{2} | 128 | 4.23 × 10^{1}  | 3.47 × 10^{-1} | 9.34 × 10^{-1} |

Table 4 shows the obtained results of the simulation carried out by implementing the Sphere function, making variations in the dimensions. Table 5 shows the results for the Ackley function.

Table 4. Experimental results: the CS, FCS 1, and FCS 2 using the sphere function.

| Sphere Function |
|-----------------|-----------------|-----------------|
| Average         | Standard Deviation |
| Dim.            | CS   | FCS1 | FCS2 | Dim.            | CS   | FCS1 | FCS2 |
| 8               | 7.91 × 10^{-46} | 3.07 × 10^{-55} | 5.18 × 10^{-92} | 8 | 1.61 × 10^{-45} | 8.86 × 10^{-85} | 8.04 × 10^{-92} |
| 16              | 6.78 × 10^{-27} | 2.65 × 10^{-48} | 4.39 × 10^{-48} | 16 | 3.13 × 10^{-27} | 1.39 × 10^{-48} | 2.76 × 10^{-48} |
| 32              | 7.01 × 10^{-15} | 2.36 × 10^{-23} | 1.31 × 10^{-20} | 32 | 2.88 × 10^{-15} | 1.18 × 10^{-23} | 6.57 × 10^{-21} |
| 64              | 6.43 × 10^{-8}  | 5.22 × 10^{-10} | 1.67 × 10^{-7}  | 64 | 2.01 × 10^{-8}  | 1.94 × 10^{-10} | 4.38 × 10^{-8} |
| 128             | 8.33 × 10^{-4}  | 1.54 × 10^{-3}  | 3.33 × 10^{-2}  | 128 | 1.54 × 10^{-4}  | 3.19 × 10^{-4}  | 5.79 × 10^{-3} |

Table 5. Results: the CS, FCS1, and FCS2 using the Ackley function.

| Ackley Function |
|-----------------|-----------------|-----------------|
| Average         | Standard Deviation |
| Dim.            | CS   | FCS1 | FCS2 | Dim.            | CS   | FCS1 | FCS2 |
| 8               | 7.92 × 10^{-15} | 1.95 × 10^{-15} | 2.31 × 10^{-15} | 8 | 3.25 × 10^{-15} | 1.64 × 10^{-15} | 1.76 × 10^{-15} |
| 16              | 2.63 × 10^{-7}  | 4.44 × 10^{-15} | 4.44 × 10^{-15} | 16 | 2.99 × 10^{-7}  | 0.00 × 10^{0}   | 0.00 × 10^{0}   |
| 32              | 1.29 × 10^{-3}  | 3.27 × 10^{-11} | 7.74 × 10^{-10} | 32 | 1.12 × 10^{-3}  | 1.09 × 10^{-11} | 1.81 × 10^{-10} |
| 64              | 2.11 × 10^{-1}  | 1.06 × 10^{-4}  | 1.73 × 10^{-3}  | 64 | 2.71 × 10^{-1}  | 1.95 × 10^{-3}  | 2.44 × 10^{-4}  |
| 128             | 2.79 × 10^{0}   | 2.00 × 10^{-1}  | 1.55 × 10^{0}   | 128 | 2.52 × 10^{-1}  | 3.20 × 10^{-2}  | 1.90 × 10^{-1}  |

In Table 6, the obtained results can be seen, both the average and standard deviation of the Rastrigin function.
Table 6. Results: the CS, FCS1, and FCS2 using the Rastrigin function.

| Dim. | CS     | FCS1   | FCS2   | Dim. | CS     | FCS1   | FCS2   |
|------|--------|--------|--------|------|--------|--------|--------|
| 8    | 1.08 × 10^{-2} | 0.00 × 10^{0} | 0.00 × 10^{0} | 8    | 4.06 × 10^{-2} | 0.00 × 10^{0} | 0.00 × 10^{0} |
| 16   | 1.63 × 10^{1}   | 9.14 × 10^{0}   | 1.00 × 10^{0}   | 16   | 2.35 × 10^{1}   | 3.39 × 10^{0}   | 3.17 × 10^{0}   |
| 32   | 7.83 × 10^{1}   | 7.66 × 10^{1}   | 7.99 × 10^{0}   | 32   | 1.02 × 10^{1}   | 1.70 × 10^{0}   | 1.88 × 10^{0}   |
| 64   | 2.67 × 10^{2}   | 3.05 × 10^{0}   | 3.38 × 10^{0}   | 64   | 2.33 × 10^{1}   | 4.80 × 10^{0}   | 4.58 × 10^{0}   |
| 128  | 7.72 × 10^{2}   | 8.91 × 10^{2}   | 9.49 × 10^{2}   | 128  | 4.16 × 10^{1}   | 6.91 × 10^{1}   | 6.58 × 10^{1}   |

The experimental results for the Griewank function are shown in Table 7.

Table 7. Results: the CS, FCS1, and FCS2 using the Griewank function.

| Dim. | CS     | FCS1   | FCS2   | Dim. | CS     | FCS1   | FCS2   |
|------|--------|--------|--------|------|--------|--------|--------|
| 8    | 2.84 × 10^{-2} | 1.76 × 10^{-2} | 1.35 × 10^{-2} | 8    | 9.53 × 10^{-3} | 8.12 × 10^{-3} | 7.88 × 10^{-3} |
| 16   | 5.01 × 10^{-7} | 8.66 × 10^{-7} | 5.29 × 10^{-7} | 16   | 3.39 × 10^{-6} | 5.21 × 10^{-6} | 8.41 × 10^{-13} |
| 32   | 8.76 × 10^{-8} | 3.24 × 10^{-15} | 3.98 × 10^{-9} | 32   | 1.11 × 10^{-7} | 6.85 × 10^{-15} | 1.80 × 10^{-8} |
| 64   | 3.57 × 10^{-4} | 1.98 × 10^{-6} | 2.74 × 10^{-4} | 64   | 9.93 × 10^{-4} | 1.71 × 10^{-6} | 1.41 × 10^{-4} |
| 128  | 1.72 × 10^{-1} | 2.90 × 10^{-1} | 1.12 × 10^{0}  | 128  | 3.56 × 10^{-2} | 5.09 × 10^{-2} | 2.51 × 10^{-2} |

In Tables 2–6, the experiments were carried out with test mathematical functions, varying from 8, 16, 32, 64, and 128 dimensions, in order to analyze the behavior of the algorithm with respect to the original algorithm and demonstrate that the results can be optimized by implementing type-1 and type-2 fuzzy logic.

7. Conclusions

In the present work, we visualize the results of CS implementing type-1 and type-2 fuzzy logic. The tests were carried out with benchmark mathematical functions, showing that for all the dimensions, our proposals, with respect to the average results, show that the FCS1 and FCS2 algorithms are better than the original CS in the Rosenbrock, sphere, Griewank, and Ackley functions. The Rastrigin function is a complex function for the proposal, for which further study is required to find alternatives for the search space range, and iterations to analyze its behavior. We have to mention that the membership functions that the fuzzy system uses are symmetrically defined for reducing the search space of parameter values, which helps the efficiency of the proposed parameter adaptation. We have to mention that one limitation of our study is that the type-2 fuzzy CS variant is slower than the type-1 variant and the original CS, but results are better. In any case, the optimization is performed offline for designing the optimal architectures or topologies of intelligent systems; this will not be a significant problem.

As a future work, we will analyze the behavior of the FOU (footprint uncertainty) to evaluate its behavior when increasing or decreasing the FOU size. We believe that finding the optimal FOU will improve the results even more. Moreover, we plan to implement generalized type 2 fuzzy logic for improving CS, but it is important to find the ideal values of the parameters to be able to consider generalized type 2 fuzzy logic. In addition, we plan to optimize fuzzy controllers with the fuzzy variants of CS proposed in this paper. This is intended to contribute to the optimization of the bio-inspired search algorithm of cuckoo search with Lévy flights. Finally, we could apply the proposed optimization to other problems, such as in time series prediction [45].
Author Contributions: Conceptualization, O.C. and F.V.; methodology, O.C.; software, M.G.; validation, M.G., F.V. and O.C.; formal analysis, O.C.; investigation, M.G.; resources, O.C.; writing—original draft preparation, M.G.; writing—review and editing, O.C.; supervision, F.V.; project administration, O.C.; funding acquisition, F.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zadeh, L.A. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. 1978, 1, 3–28. [CrossRef]
2. Zadeh, L.A.; Aliev, R.A. Fuzzy Logic Theory and Applications: Part I and Part II; World Scientific Publishing: Singapore, 2018.
3. Bhardawaj, S.; Sharma, R.C.; Sharma, S.K. Development of multibody dynamical using MR damper based semi-active bio-inspired chaotic fruit fly and fuzzy logic hybrid suspension control for rail vehicle system. Proc. Inst. Mech. Eng. Part K. J. Multi-Body Dyn. 2020, 234, 723–744. [CrossRef]
4. Bernal, E.; Castillo, O.; Soria, J.; Valdez, F. Optimization of fuzzy controller using galactic swarm optimization with type-2 fuzzy dynamic parameter adjustment. Axioms 2019, 8, 26. [CrossRef]
5. Valdez, F. A review of optimization swarm intelligence-inspired algorithms with type-2 fuzzy logic parameter adaptation. Soft Comput. 2020, 24, 215–226. [CrossRef]
6. Mancilla, A.; Castillo, O.; Valdez, M.G. Optimization of fuzzy logic controllers with distributed bio-inspired algorithms. In Recent Advances of Hybrid Intelligent Systems Based on Soft Computing; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–11.
7. Mudi, R.K.; Dey, C. Nature-inspired and hybrid optimization algorithms on interval Type-2 fuzzy controller for servo processes: A comparative performance study. SN Appl. Sci. 2020, 2, 1292.
8. Valdez, F.; Castillo, O.; Melin, P. Bio-inspired algorithms and its applications for optimization in fuzzy clustering. Algorithms 2021, 14, 122. [CrossRef]
9. Tamir, D.E.; Rishe, N.D.; Kandel, A. Fifty Years of Fuzzy Logic and Its Applications; Springer: Berlin/Heidelberg, Germany, 2015; Volume 326.
10. Miramontes, I.; Melin, P.; Prado-Arcehiga, G. Comparative study of bio-inspired algorithms applied in the optimization of fuzzy systems. In Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine; Springer: Berlin/Heidelberg, Germany, 2020; pp. 219–231.
11. Olivas, F.; Amador-Angulo, L.; Perez, J.; Caraveo, C.; Valdez, F.; Castillo, O. Comparative Study of Type-2 Fuzzy Particle Swarm, Bee Colony and Bat Algorithms in Optimization of Fuzzy Controllers. Algorithms 2017, 10, 101. [CrossRef]
12. Santillan, J.H.; Tapucar, S.; Manliguez, C.; Calag, V. Cuckoo search via Levy flights for the capacitated vehicle routing problem. J. Ind. Eng. Int. 2018, 14, 293–304. [CrossRef]
13. Kishnani, M.; Pareek, S.; Gupta, R. Optimal tuning of PID controller by cuckoo search via Levy flights. In Proceedings of the 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), Unnao, India, 1–2 August 2014.
14. Zhang, H.; Cui, G. Optimal heat exchanger network synthesis based on improved cuckoo search via Levy flights. Chem. Eng. Res. Des. 2018, 134, 62–79. [CrossRef]
15. Chakraborty, S.; Mali, K. Biomedical image segmentation using fuzzy multilevel soft thresholding system coupled modified cuckoo search. Biomed. Signal Process. Control 2022, 72, 103324. [CrossRef]
16. Subramani, B.; Veluchamy, M. Cuckoo search optimization-based image color and detail enhancement for contrast distorted images. Color Res. Appl. 2022. [CrossRef]
17. Ochoa-Zezzatti, A.; Castillo, O.; Melin, P.; Castillo, N.; Bustillos, S.; Arreola, J. Shipwrecked on fear: Selection of electives in school minorities in a university using cuckoo search algorithm. In Recent Advances on Hybrid Approaches for Designing Intelligent Systems; Springer: Berlin/Heidelberg, Germany, 2014; pp. 139–150.
18. Fan, J.; Xu, W.; Huang, Y.; Samuel, R.D.J. Application of Chaos Cuckoo Search Algorithm in computer vision technology. Soft Comput. 2021, 25, 12373–12387. [CrossRef]
19. Jiao, W.; Chen, W.; Zhang, J. An improved cuckoo search algorithm for multithreshold image segmentation. Secur. Commun. Netw. 2021, 2021, 6036410. [CrossRef][PubMed]
20. Yousri, D.; Elaziz, M.A.; Abualigah, L.; Oliva, D.; Al-Qaness, M.A.; Ewees, A.A. COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. Appl. Soft Comput. 2021, 101, 107052. [CrossRef][PubMed]
21. Shen, D.; Ming, W.; Ren, X.; Xie, Z.; Zhang, Y.; Liu, X. A Cuckoo Search Algorithm Using Improved Beta Distributing and Its Application in the Process of EDM. Crystals 2021, 11, 916. [CrossRef]
22. Wang, G.-G.; Deb, S.; Gandomi, A.H.; Zhang, Z.; Alavi, A.H. Chaotic cuckoo search. Soft Comput. 2016, 20, 3349–3362. [CrossRef]
23. Tsipianitis, A.; Tsompanakis, Y. Improved Cuckoo Search algorithmic variants for constrained nonlinear optimization. Adv. Eng. Softw. 2020, 149, 102865. [CrossRef]
24. Afzal, A.; Samee, A.M.; Jilte, R.; Islam, T.; Manokar, A.M.; Razak, K.A. Battery thermal management: An optimization study of parallelized conjugate numerical analysis using Cuckoo search and Artificial bee colony algorithm. *Int. J. Heat Mass Transf.* 2021, 166, 120798. [CrossRef]

25. Ou, X.; Wu, M.; Pu, Y.; Tu, B.; Zhang, G.; Xu, Z. Cuckoo search algorithm with fuzzy logic and Gauss-Cauchy for minimizing localization error of WSN. *Appl. Soft Comput.* 2022, 109211.

26. Liang, Q.; Mendel, J.M. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Trans. Fuzzy Syst.* 2000, 8, 535–550. [CrossRef]

27. Mendel, J.; Hagras, H.; Tan, W.W.; Melek, W.W.; Ying, H. *Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2014.

28. Liu, X.; Lin, Y.; Wan, S.-P. New efficient algorithms for the centroid of an interval type-2 fuzzy set. *Inf. Sci.* 2021, 570, 468–486. [CrossRef]

29. Castillo, O.; Melin, P. A review on interval type-2 fuzzy logic applications in intelligent control. *Inf. Sci.* 2014, 279, 615–631. [CrossRef]

30. Castillo, O.; Melin, P. A review on the design and optimization of interval type-2 fuzzy controllers. *Appl. Soft Comput.* 2012, 12, 1267–1278. [CrossRef]

31. Castillo, O.; Castro, J.R.; Melin, P. Type-2 fuzzy logic systems. In *Interval Type-3 Fuzzy Systems: Theory and Design*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 5–11.

32. Butt, M.A.; Akram, M. A novel fuzzy decision-making system for CPU scheduling algorithm. *Neural Comput. Appl.* 2016, 27, 1927–1939. [CrossRef]

33. Akram, M.; Shahzad, S.; Butt, A.; Khaliq, A. Intuitionistic fuzzy logic control for heater fans. *Math. Comput. Sci.* 2013, 7, 367–378. [CrossRef]

34. Habib, S.; Akram, M.; Ashraf, A. Fuzzy climate decision support systems for tomatoes in high tunnels. *Int. J. Fuzzy Syst.* 2017, 19, 751–775. [CrossRef]

35. Ashraf, A.; Akram, M.; Sarwar, M. Fuzzy decision support system for fertilizer. *Neural Comput. Appl.* 2014, 25, 1495–1505. [CrossRef]

36. Mutovkina, N.; Borodulin, A. Diagnosis of Atypical Forms of Myocardial Infarction Based on Fuzzy Logic. In *The International Conference on Artificial Intelligence and Logistics Engineering*; Springer: Berlin/Heidelberg, Germany, 2022.

37. Wanti, L.P.; Puspitasari, L. Optimization of the Fuzzy Logic Method for Autism Spectrum Disorder Diagnosis. *J. RESTI (Rekayasa Sist. Dan Teknol. Inf.)* 2022, 6, 16–24. [CrossRef]

38. Ahmed, T.I.; Bhola, J.; Shabaz, M.; Singla, J.; Rakhra, M.; More, S.; Samori, I.A. Fuzzy logic-based systems for the diagnosis of chronic kidney disease. *BioMed Res. Int.* 2022, 2022, 2653665.

39. Yang, X.-S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009.

40. Guerrero-Luis, M.; Valdez, F.; Castillo, O. A Review on the Cuckoo Search Algorithm. In *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 113–124.

41. Guerrero, M.; Castillo, O.; García, M. Cuckoo search via lévy flights and a comparison with genetic algorithms. In *Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 91–103.

42. Guerrero, M.; Castillo, O.; García, M. Study of Parameter Variations in the Cuckoo Search Algorithm and the Influence in Its Behavior. In *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*; Melin, P., Castillo, O., Kacprzyk, J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 199–210.

43. Guerrero-Luis, M.; Valdez, F.; Castillo, O. A Review on the Cuckoo Search Algorithm. In *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 113–124.

44. Yang, J.-Q.; Chen, C.-H.; Li, J.-Y.; Liu, D.; Li, T.; Zhan, Z.-H. Compressed-Encoding Particle Swarm Optimization with Fuzzy Learning for Large-Scale Feature Selection. *Symmetry* 2022, 14, 1142. [CrossRef]

45. Castillo, O.; Castro, J.R.; Melin, P.; Rodríguez-Díaz, A. Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction. *Soft Comput.* 2014, 18, 1213–1224. [CrossRef]