Journal of Universal Mathematics
Vol.1 No.1 pp.32-61 (2018)
ISSN-2618-5056

A NEW ALGORITHM BASED ON THE CUCKOO SEARCH WITH DYNAMIC ADAPTATION OF PARAMETERS USING FUZZY SYSTEMS

MARIBEL GUERRERO, MARIO GARCÍA, JOSE SORIA, OSCAR CASTILLO*

ABSTRACT. In this work we studied of the parameters of the Cuckoo Search Algorithm via Levy Flights (CS). The main goal of the paper is designing a novel hybrid approach for modifying the Cuckoo Search Algorithm using a Fuzzy Inference System of the Mamdani type for calculating the optimal parameter values independent of the benchmark problem, which we are calling Fuzzy Cuckoo Search (for its acronyms FCS). In this paper different variants of the FCS are presented and the difference is the number of parameters adjusted by the fuzzy control system and the number of rules. Results show that the FCS outperforms the original version of the CS algorithms and the OCS variant of the algorithm proposed by Zhao. The statistical test shows that using a type - 1 fuzzy system in conjunction with the cuckoo search algorithms provides the best solutions

1. INTRODUCTION

In this paper a fuzzy logic based approach for enhancing the Cuckoo search (CS) algorithm is presented. The main idea is to use fuzzy systems to dynamically adapt the relevant parameters of the CS algorithm during execution. Mathematical functions are used to evaluate the performance of the algorithms with the intention of later considering engineering problems and also to facilitate the analysis. However throughout our research we found that there are complex mathematical functions.

The main contribution of this paper is proposal a novel hybrid approach for modifying the Cuckoo Search Algorithm using a Fuzzy Inference System of the Mamdani type for calculating the optimal parameter values independent of the benchmark problem, and this is currently not found in literature an implementation of dynamic adjustment of parameters using a fuzzy inference system to help the convergence.

In this work we analyzed several proposals for a FCS to determine which parameters help produce better results using fuzzy systems, if individually or in combination parameters and comparison with the original CS algorithm and the OCS variant.

Talking about fuzzy logic this transport us to the year 1965 at the University of California Berkeley and to mention who is known as the father of Fuzzy Logic [27].

Zadeh introduced the concept of a fuzzy set [28], which is based on the notion that the elements on which human thinking is build are not numbers but linguistic labels. Fuzzy logic can represent the common knowledge as a form of language mostly qualitative and that is not necessarily a quantity in a mathematical language [20,31].

On the other hand, in many problems, fuzzy systems have been proved to be efficient tools and shown excellent ability when describing nonlinear systems, because fuzzy models are capable of handling uncertainties, such as the nonlinearity and ambiguity involved in a real system [4,7].

Fuzzy systems have demonstrated their ability as system identification tools: for example in [15] the authors propose an improvement to the convergence and diversity of the swarm in PSO using Fuzzy Logic, and the results show that the proposed approach improves the performance of the original PSO.

In another work a design procedure for a Mamdani fuzzy logic controller is presented, including its rule base minimization. A genetic algorithm is used for finding stabilizing controllers that minimize the number of rules [1].

Another paper concentrated on the design of a minimum rule based fuzzy-logic controller for robot navigation, and obstacle avoidance in clusters environment, based on the Mamdani type fuzzy method. The success of mobile robot navigation control depends mostly on the accuracy of absolute measurements of its position: obstacle distances, goal distance, velocity, orientation, and its rate of change its heading angle. The whole navigation system has been tested in a simulation environment with satisfying results [17].

In another paper, they present a new technique using a modified measure and blending of cuckoo search and particle swarm optimization for low contrast images to enhance image adaptively. Image enhancement is an important procedure of image processing and analysis. Experimental results demonstrate that the proposed method is robust, adaptive and exhibits better performance than other methods mentioned in [14].

The Cuckoo Search algorithm via Levy flights, is a meta-heuristic optimization method proposed by [24].

CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights. CS was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds [23].

CS is relatively young, has endless applications and variants, the intent of this paper is to help the algorithm to improve its convergence using fuzzy logic.

The rest of the paper is organized as follows: Section 2 presents an overview of the Cuckoo Search Algorithm via Lévy Flights. Section 3 presents a new CS method with dynamic adjustment of parameters using fuzzy logic. In Section 4 we present a Study of parameters variations in the CS Algorithm. In Section 5 the Benchmark functions are defined, and in Section 6 we present different models of Fuzzy Cuckoo Search. In Section 7 we present our proposal. Section 8 shows the simulation results. In Section 9 several Statistical tests are presented. Finally, Conclusions are offered in Section 10.

2. CUCKOO SEARCH ALGORITHM

The CS algorithm is inspired by the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not its own and either destroy the egg or abandon the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds [24].
There are different variants of CS algorithms in line with this work [11].
For example, some original studies in this area are:
Cuckoo search via Lévy flights [11].
An efficient cuckoo search algorithm for numerical function optimization [16].
Multimodal function optimization[12].
In this paper we consider as a basis for proposing the new approach the Cuckoo Search via Lévy flights algorithm using one objective.

2.1 EQUATIONS TO GENERATE NEW POSITIONS

When generating new solutions $x_i^{(t+1)}$, for say a cuckoo i, a Lévy flight is performed using Equation 1:

$$x_i^{(t+1)} = x_i^t + \alpha \times randn(size(D)) \oplus Lévy(\beta) \oplus S_i \tag{1}$$

Where:

$x_i^{(t+1)}$: is the vector representing the new position.

x_i^t : is the vector representing current position.

α : is the step size, which should be related to the scales of the problem of interest, where $\alpha \geq 0$.

$randn$ (size (D)): is a random scalar value.

The product $\oplus$ means entry-wise multiplications.

$\text{Lévy}(\beta)$: it is a Lévy flight.

In Equation 2 we have that S_i is defined by the following difference:

$$S_i = (x_i^t - x_{best}) \tag{2}$$

Where:

x_i^t : is the position vector in the nest we are evaluating

x_{best} : is a vector containing the best position found so far

If $x_i^t = x_{best}$ the solution is maintained in the process

2.2 EQUATION OF THE LÉVY FLIGHT

The global random walk is carried out by using Levy flights. The main equation of the Levy flight is:

$$\text{Lévy}(\beta) = \frac{\sigma(\beta) \times \text{randn}[D]}{\left|\text{randn}[D]\right|^{\frac{1}{\beta}}} \tag{3}$$

Where:

β : is a constant $(1 < \beta \leq 2)$

The randn[D] function generates a uniform integer in the interval [1, D].

The standard deviation sigma is calculated with the following equation:

$$\sigma_i(\beta) = \left\{ \frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma\left[\frac{(1+\beta)}{2}\right] \times \beta 2^{\left(\frac{\beta-1}{2}\right)}} \right\}^{\frac{1}{\beta}}, \tag{4}$$

The Gamma function is defined by the integral:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt \tag{5}$$

The Lévy flight represents a random trajectory and it is the length of the step obtained by means of a Lévy distribution (Equation 6):

$$\text{Lévy} \sim u = t^{-\beta}, \qquad (1 < \beta < 2) \tag{6}$$

Which has an infinite variance with an infinite mean.

2.3 LEVY FLIGHTS

Many animals and insects have demonstrated the typical characteristics of Lévy flights [2,3,5,18,21.

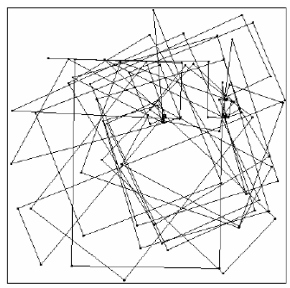


Figure 1. Possible Lévy flight path.

A recent study by Reynolds and Frye shows that fruit flies or Drosophila melanogaster; explore their landscape using a series of straight flight paths punctuated by a sudden 90° turn, leading to a Lévy-flight-style intermittent scale free search pattern [3]. In Figure 1, we can observe an illustration of the behavior of a Lévy flight.

2.4 BASIC RULES IN THE CS ALGORITHMS

The breeding behavior of cuckoos is described below in the basic steps of cuckoo search using the following three rules [23,24]:
Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.
The best nests with high quality of eggs will carry over to the next iterations.

The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability *Pa* ε [0, 1]. In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest. For simplicity, this last assumption can be approximated by the fraction *Pa* of the *n* nests that are replaced by new nests (with new random solutions).

Each egg in a nest represents a solution and the cuckoo egg represents a new solution. Therefore, there is no difference between an egg, a nest, and a solution.

2.5 PSEUDOCODE

The pseudocode of the Cuckoo Search is presented in the CS Algorithm [24]:

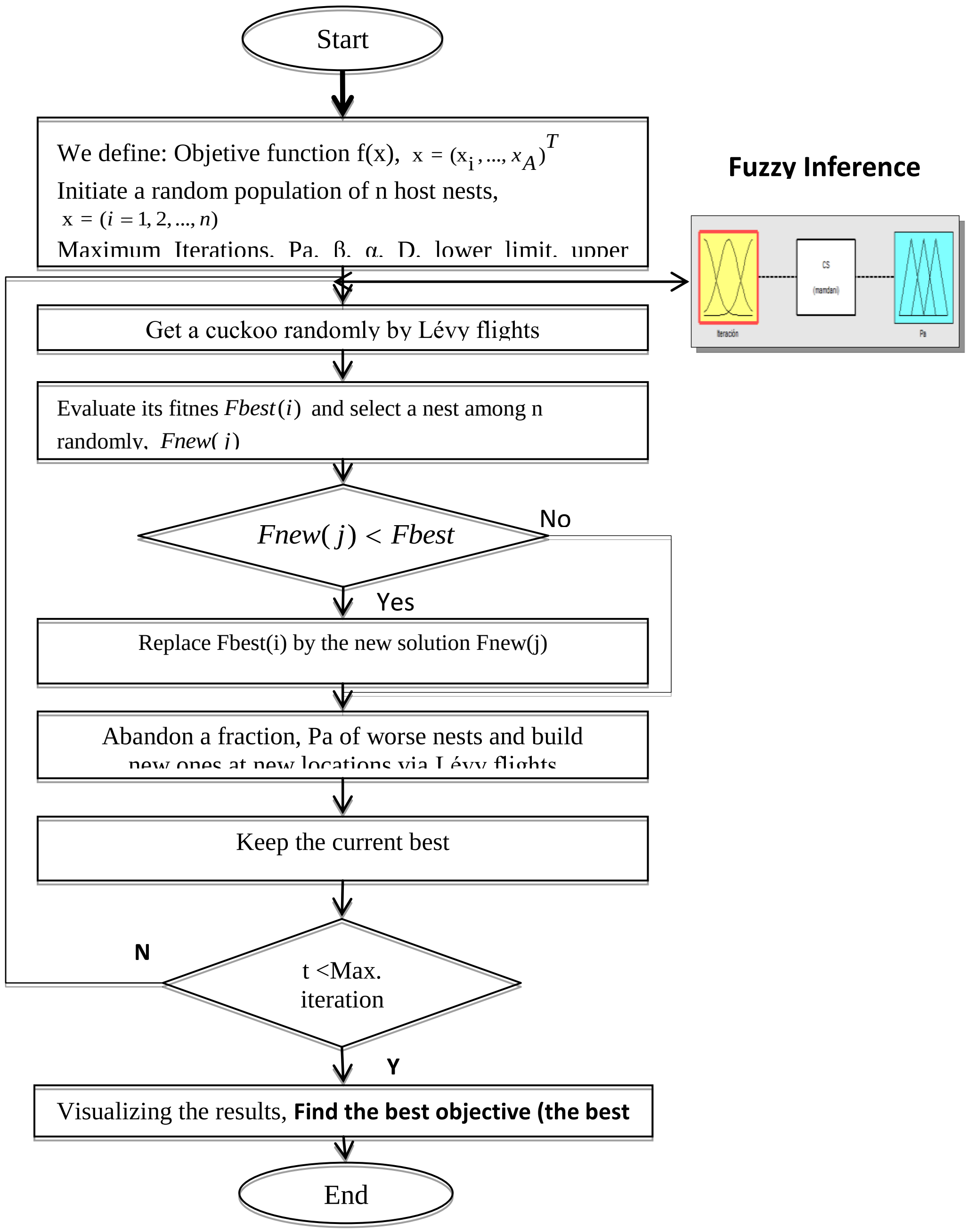
1. Begin
2. Objective function f(x), x= (〖x_1,…,x_d)〗 ^T
3. Generate initial population of n host nests x_i (i = 1, 2, ..., n)
4. while (t <Maximum iteration) or (stop criterion)
5. Get a cuckoo randomly by Lévy flights evaluate its quality/fitness Fi
6. Choose a nest among n (say, j) randomly
7. if (Fi >Fj),
8. replace j by the new solution;
9. end if
10. A fraction (Pa) of the worse nests is abandoned and new ones are built;
11. Keep the best solutions (or nests with quality solutions);
12. Rank the solutions and find the current best
13. end while
14. Postprocess results and visualization
15. End

2.6 HYBRID CUCKOO SEARCH

The hybrid algorithms combine algorithms in order to help convergence. It is another way of improving convergence. We mention below some proposed hybrid approaches.

Hybrid CS/GA by Ghodrati and Lotfi: This paper presents a hybrid approach of Cuckoo Search (CS) and Genetic Algorithm (GA), in this case each cuckoo lays one egg at a time, but in the proposed hybrid algorithm, in order to lay more eggs they used the genetic algorithms strategy (Crossover) for their reproduction [9].

Hybrid CS by Li and Yin [14]: Cuckoo Search can deal with multimodal problems naturally and efficiently. However, researchers have also attempted to improve its efficiency further so as to obtained better solutions than those in the literature [6].



These variants of CS have applications in the design engineering and other applications [13,26]

3. NEW CS ALGORITHM WITH DYNAMIC ADJUSTMENT OF PARAMETERS USING FUZZY LOGIC

In this Section we present the main idea of the proposed approach, which is a new algorithm, which we have called Fuzzy Cuckoo Search (FCS).

3.1. STRUCTURE OF THE FUZZY INFERENCE SYSTEM

The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of fuzzy rules, a database (or dictionary) which defines the membership functions used in the rules, and a reasoning mechanism, which performs the inference procedure (usually called fuzzy reasoning), [19].

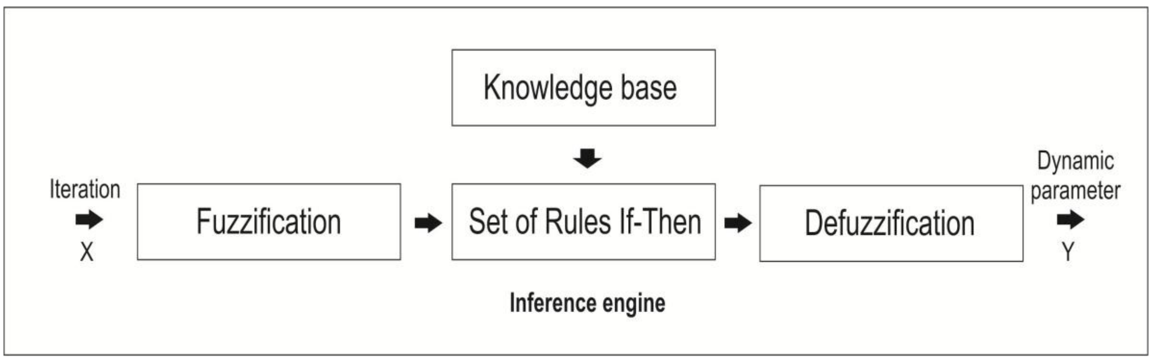


Figure 2. Block diagram of the Fuzzy Inference System

Figure 2 illustrated the structure of a fuzzy inference system, where the "input" of the FIS in our approad is the iteration and the "output" is the dynamic adjustment of parameters.

3.2. FLOW CHART OF FUZZY CUCKOO SEARCH

Figure 3 shows a flow chart that illustrates the sequence of processes and decisions that the CS algorithm performs and the intervention of a Fuzzy Inference System in the dynamic parameter setting. In this paper we are presenting different new Fuzzy Cuckoo Search (FCS) variants with dynamic adjustment of parameters applied to benchmark functions.

Figure 3. General proposed Fuzzy Cuckoo Search algorithm.

4 STUDY OF THE VARIATION OF PARAMETER PA MANUALLY IN THE CS ALGORITHM

In this section we are presenting an analysis and the selection of the parameters to be adjusted and to be considered as dynamic parameters in CS. Basically we manually change the values of the different parameters to analyze their effect on the performance of the CS algorithm.

The design of an optimal fuzzy inference system is not easy, beginning by defining the number of antecedents and consequents, the type and number of membership functions, and the complexity increases when trying to select the ideal parameters and number of rules to achieve the optimal result. In this case we need to analyze the relevant variables to be considered in the fuzzy system.

For this reason, we decided to study the behavior of the parameters initially with the Rosenbrock function, because in previous simulation results with CS this function had the worse outcomes [10].

4.1 DESCRIPTION OF THE PARAMETERS OF THE CS ALGORITHM

The control parameters for the CS algorithm are:

- Discovering probability (*Pa*). Pa is a very important parameter because the choice of solutions is carried out using this parameter, where good solutions will advance to the next iteration and the not so good are replaced by new solutions. The value for Pa ϵ [0, 1].

- The scale factor (β). This parameter is implicit in Lévy flights and is used to explore and exploit the search space. The value is $(1 < \beta \leq 2)$.

- And the convergence speed (α) the value is α ϵ [0, 1].

4.2 PARAMETERS FOR THE SIMULATION

The Rosenbrock function was evaluated with 32 dimensions. The mathematical expression is defined below.

$$f(x)=\sum_{j=1}^{n_z/2}[100(x_{2j}-x_{2j-1}^2)^2+(1-x_{2j-1})^2] \tag{7}$$

With $x_j \in [-5,10]$ and $f^*(x) = 0.0$

In Table 1 we present the values implemented for each of the parameters of the simulations.

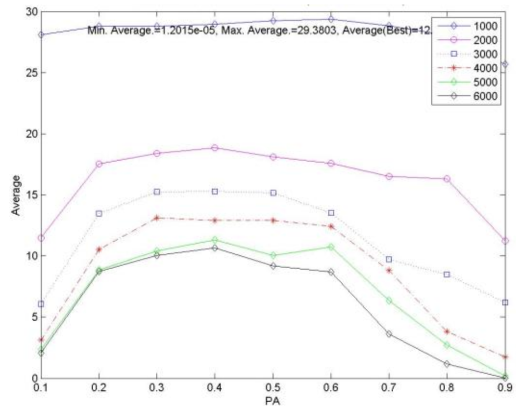
Table 1. Parameters used in the simulations

Parameters	*Pa*	**β**	***α***
Number of simulations	30	30	30
Number of nests	100	100	100
Number of iterations is variation	1000 to 6000	1000 to 6000	1000 to 6000
Pa	0.1 to 0.9	0.75	0.75
α	0.05	0.05	[0.1 to 0.9] and [1.1 to 1.9]
β	1.7	1.1 to 1.9	1.7
Dimensions	32	32	32

In Graphs 1 to 3 we present in the X axis the average results of a set of 30 simulations and on the Y axis the corresponding parameter considered in the study (Pa, α , β). Each point on the graph represents the particular average according to the iteration and the value of the parameter currently being evaluated.

4.3 VARYING THE PA PARAMETER

The egg laid by a cuckoo is discovered by the host bird with a probability Pa ϵ [0, 1]. In this case, the bird can either get rid of the eggs, or simply abandon the nest and build a complexly new nest.



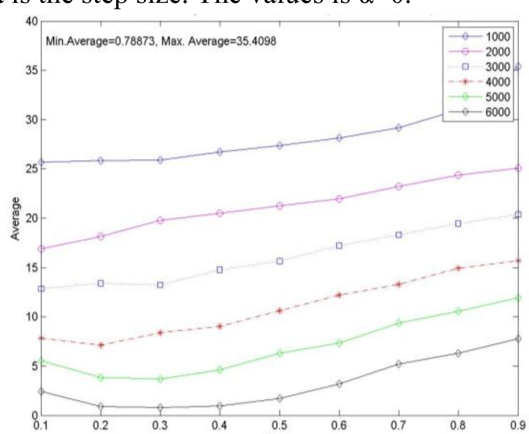
Graph. 1. Variation of Pa - Rosenbrock Function (1000-6000 Iterations)

Graph 1 allows us to visualize the behavior of the algorithm as iterations increase; the most important thing is to find a pattern that is repeated throughout the iterations to determine the appropriate range to consider in the implementation of the fuzzy inference system.

The pattern tends to show good results in the CS algorithm are in the range of Pa ϵ [0.7 to 0.9].

4.4 VARYING THE PARAMETER ALPHA

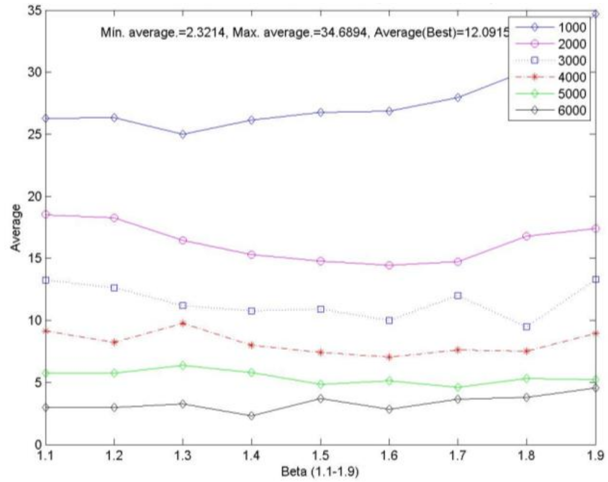
The parameter Alpha is the step size. The values is $\alpha > 0$.



Graph. 2. Variation of α - Rosenbrock function (1000-6000 iterations).

4.5 VARYING THE BETA PARAMETER

This parameter is implicit in Lévy flights and is used to explore and exploit the search space. The value is $(1 < \beta \leq 2)$.



Graph. 3. Variation of β-Rosenbrock Function (1000-6000 iterations)

The results in Graphs 2 and 3 do not present a clear trend for the behavior of these parameters. For this reason we continue to investigate the optimal range that helps the algorithm to improve results.

5 BENCHMARK FUNCTIONS

From a mathematical point of view, an algorithm is an iterative process, which aims to generate a new and better solution x_i^{t+1} to a given problem from the current solution x^t with respect to time t (iterations). In this case the problem is the optimization of benchmark functions.

Now we present the mathematical functions used in the simulations and a corresponding short description of the each function.

5.1 F1 Function

$$f(x) = \sum_{j=1}^{n_x} x_j^2 \tag{8}$$

Equation 8 represents a quadratic function in which we want to find the optimal value of f(x) in a d-dimensional space. Here n is the number of variables to be optimized, the objective is to find the minimum f(x*)=0, which occurs at x*=(0,0,...,0).

5.2 F2 Function

$$f(x) = 10n + \sum_{j=1}^{n_x} [x_j^2 - 10\cos(2\pi x_j)] \tag{9}$$

This function has a unique global minimum f(x*)=0 that occurs at x*=(0,0...,0) and n is the number of variables to be optimized.

5.3 F3 Function

$$f(x) = \frac{1}{4000} + \sum_{n=1}^{n_x} x_n^2 - \prod_{n=1}^{N} \cos\left(\frac{x_n}{\sqrt{1}}\right) + 1 \tag{10}$$

This F5 function has many local minima. But a single global minimum f(x*)=0 at x*=(0,0,..,0), where n represents the dimensions.

5.4 F4 Function

$$f(x) = 20 + e - 20e^{-1/5} \sqrt{\frac{1}{n} \sum_{j=1}^{n} x_j^2} - e^{\frac{1}{n}} \sum_{j=1}^{n} \cos(2\pi x_j) \tag{11}$$

With a global minimum f(x*)=0 at x*=(0,0,..,0), here n is the number of dimensions.

6 FUZZY CUCKOO SEARCH (FCS)

The Section presents the different designs that we have considered for the Fuzzy Cuckoo Search. We use fuzzy logic to dynamically adjust the *Pa* parameter in the algorithm FCS (Pa), the parameter α in the algorithm FCS (Alpha) and the parameter β in the algorithm FCS (Beta) respectively. The aim of these variants of FCS is to improve on the original Cuckoo Search algorithm.

These combinations with different output parameters in the fuzzy inference systems, result in the following two parameters adjusting algorithms with dynamic adjustment that were developed: FCS (Pa and Alpha), FCS (Pa and Beta) and FCS (Alpha and Beta). We also integrated the algorithm with the setting of the 3 output parameters, so called: FCS (Pa, Alpha and Beta).

6.1 PROPOSED GENERAL ARCHITECTURE FOR FUZZY CUCKOO SEARCH

This Section presents a detailed description of the general architecture of the proposed modification of the original Cuckoo Search method using fuzzy logic for dynamic adjustment of parameters with the aim of helping its convergence, and for the simulation results they were tested with benchmark mathematical functions.

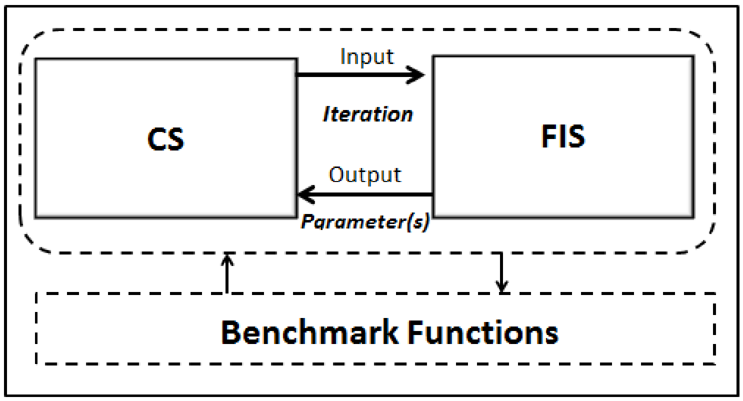


Figure 4. Our proposal for the Fuzzy Cuckoo Search (FCS)

In Figure 4 we present the proposed general architecture of FCS, with one input and one, two or three parameters as outputs for the various proposed Fuzzy Systems. The fuzzy systems are of Mamdani type and the defuzzification method is the centroid. The membership functions are of triangular form.

The same architecture was implemented in all our proposals, the changes are only being made in the combination of parameters to be adjusted.

6.2 EQUATION IMPLEMENTED IN THE FIS

Equation 12 was implemented to calculate the percentage of iterations in which we are currently, and this result is entered as an input to the FIS.

$$Iteration = \frac{Current\_Iteration}{Maximum\_Iteration} \tag{12}$$

7. THE PROPOSED FUZZY CS

In this Section we are presenting the internal structure of the fuzzy inference systems implemented in the CS algorithm.

The proposed algorithms are: FCS (Pa), FCS (Alpha), FCS (Beta), FCS (Pa and Alpha), FCS (Pa and Beta) and the integration of the three parameters FCS (Pa, Alpha and Beta).

7.1 PROPOSED FCS (PA)

In Figure 5 we can find the proposed Fuzzy Cuckoo Search with dynamic adjustment of the Pa parameter. In this Figure we are presenting the architecture of the proposed FCS (Pa) for the dynamic parameter setting of Pa, which is the probability of discovery by the host bird in its nest.

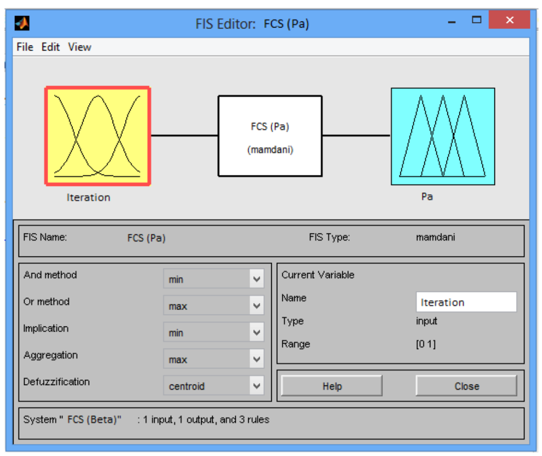


Figure 5 Proposed Fuzzy System of FSC (Pa)

The membership functions are of triangular form and in Figure 6 (A) we present the Input variable “Iteration”, which range is [0 1], and Figure 6 (B) shows the Output variable Pa with dynamic adjustment and with range of [0.7 1]. This range has better results when performing the simulations.

A) **B)**

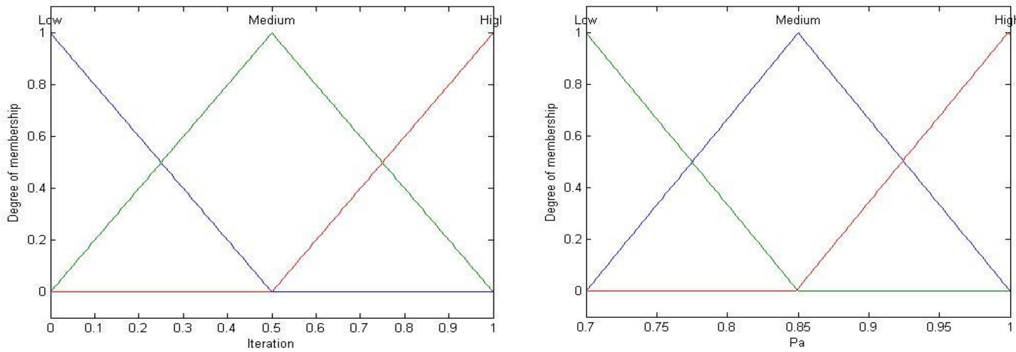


Figure 6. Fuzzy System membership functions FCS(Pa)

1. If (Iteration is Low) then (Pa is High) (1)
2. If (Iteration is Medium) then (Pa is Medium) (1)
3. If (Iteration is High) then (Pa is Low) (1)

Figure 7. Rules of operation for FCS(Pa)

In Figure 7 we are presenting the rules of the proposed fuzzy system for the Cuckoo Search with dynamic adjustment of the Pa parameter. For example: if iteration is Low then Pa is High, if iteration is Medium the Pa is Medium and if iteration is High then Pa is Low. The other proposed fuzzy systems are similar to the main fuzzy system, but the difference is in the parameter to be adjusted.

7.2 PROPOSED FCS (BETA)

In Figure 8 we show the proposal for FCS (Beta) with dynamic adjustment of the Beta parameter. The range of this variable is [1 2], and it is implemented in the expression of the Lévy distribution in Equation 6.

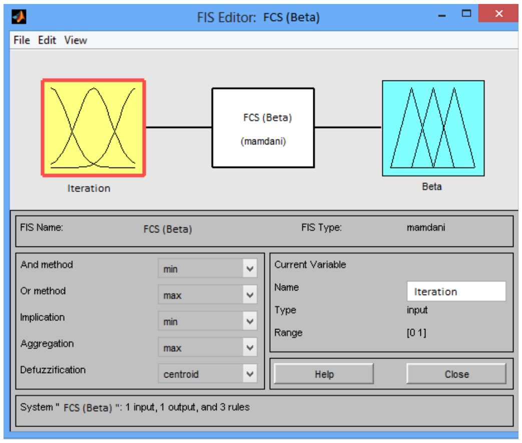


Figure 8. FCS(Beta)

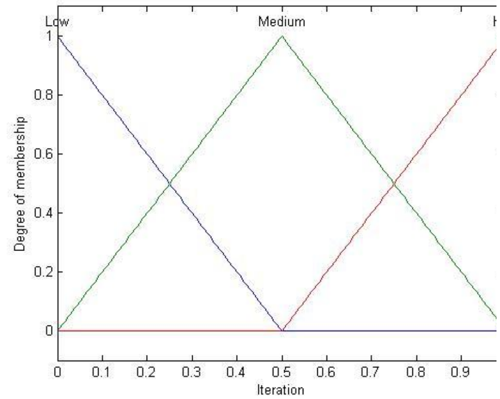


Figure 9. Input variable of FCS(Beta)

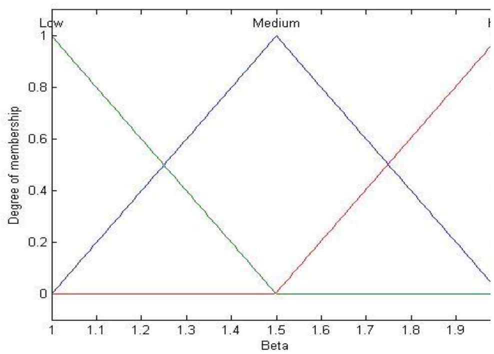


Figure 10. Output variable of FCS(Beta)

1. If (Iteration is Low) then (Beta is High) (1)
2. If (Iteration is Medium) then (Beta is Medium) (1)
3. If (Iteration is High) then (Beta is Low) (1)

Figure 11. Rules of operation for FCS (Beta)

The membership functions are of triangular form in the “Iteration” input variable and the “Beta” Output variable as are shown in Figures 9 and 10 respectively. The fuzzy system consists of 3 rules, which are presented in Figure 11.

7.3 PROPOSED FCS (ALPHA)

In Figures 12 to 15 we show the proposal for the FCS (Alpha), with dynamic adjustment of the Alpha parameter. The information about the range of this parameter from the literature is only that the alpha value should be: $\alpha \geq 0$, but for the experiments and to input this parameter to the FIS it is necessary to establish a particular range, so to perform various simulations we set the range in which the best results are found, by manually moving this parameter.

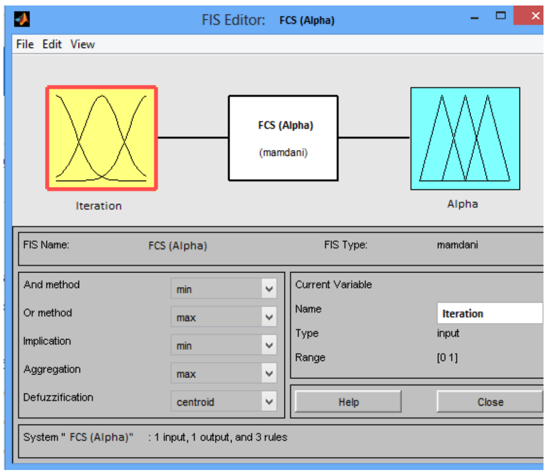


Figure 12. Proposed FCS (Alpha)

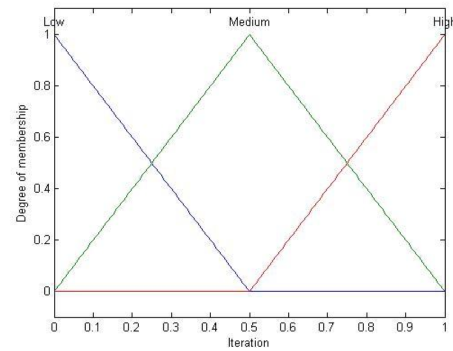


Figure 13. Input variable of FCS(Alpha)

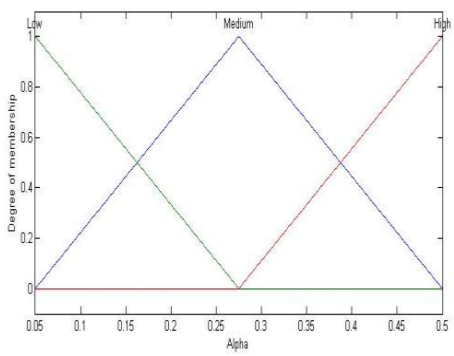


Figure 14. Output variable of FCS(Alpha)

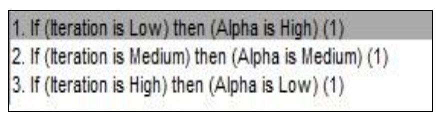

1. If (Iteration is Low) then (Alpha is High) (1)
2. If (Iteration is Medium) then (Alpha is Medium) (1)
3. If (Iteration is High) then (Alpha is Low) (1)

Figure 15. Rules of operation for FCS (Alpha)

Also, in Figure 14 the α output variable it is in a range [0.05 to 0.5]. Parameter α is implemented in Equation 1 for the discovery of new solutions (nest), and the rules of operation for the fuzzy system are presented in Figure 15.

7.4 PROPOSAL FOR THE FCS (PA AND BETA)

In Figure 16 we present the basic idea of the FCS (Pa and Beta) to combine the dynamic adjustment of the Pa and Beta parameters using a Fuzzy System for decision making.

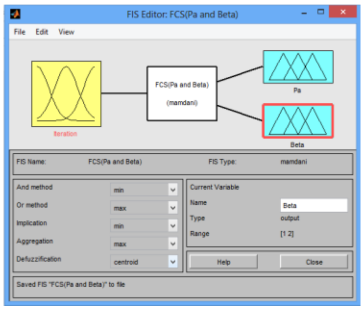


Figure 16. Proposal FCS (Pa and Beta)

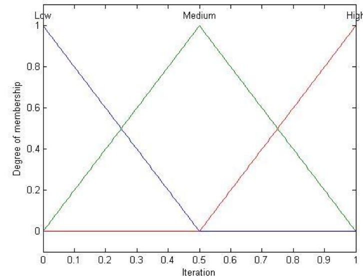


Figure 17. Input variable of FCS(Pa and Beta)

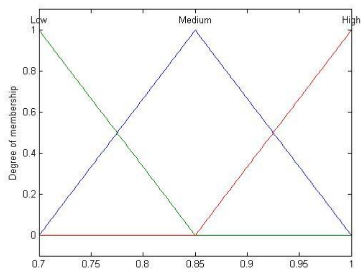


Figure 18. Output variable Pa of FCS(Pa and Beta)

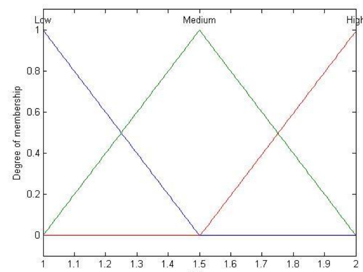


Figure 19. Output variable Beta of FCS(Pa and Beta)

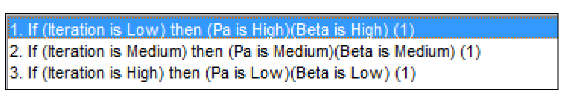

1. If (Iteration is Low) then (Pa is High)(Beta is High) (1)
2. If (Iteration is Medium) then (Pa is Medium)(Beta is Medium) (1)
3. If (Iteration is High) then (Pa is Low)(Beta is Low) (1)

Figure 20. Rules of operation for FCS (Pa and Beta)

Figures 17 to 19 illustrate the structure of the FIS, which include the “Iteration" input variable and two outputs with dynamic adjustment of the “Pa” and “Beta” parameters.

In Figure 20 we present the rules of operation for the FCS (Pa and Beta) for example: If Iteration is Low, then Pa is High and Beta is High.

In the other proposals we implement different variants with other possible combinations of parameters.

7.5 PROPOSAL FOR FCS (PA AND ALPHA)

In this proposal we present a fuzzy system with dynamic adjustment of the Pa and Alpha parameters, and in Figure 21 we present another variant of FCS (Pa and Alpha).

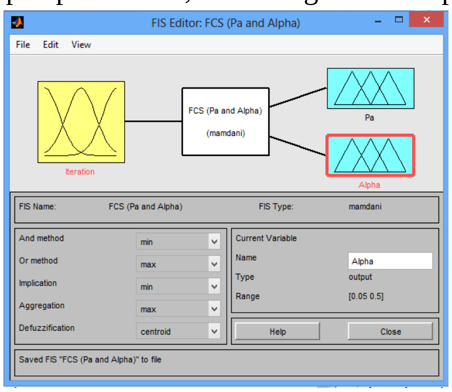


Figure 21. Proposal of FCS (Pa and Alpha)

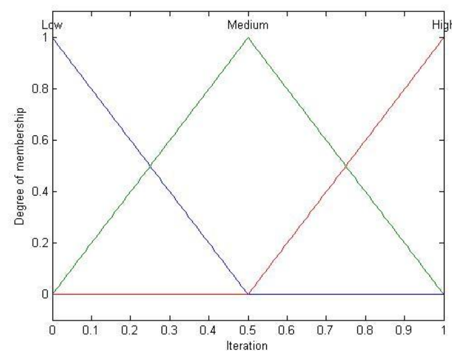


Figure 22. Input variable of the FCS(Pa and Alpha)

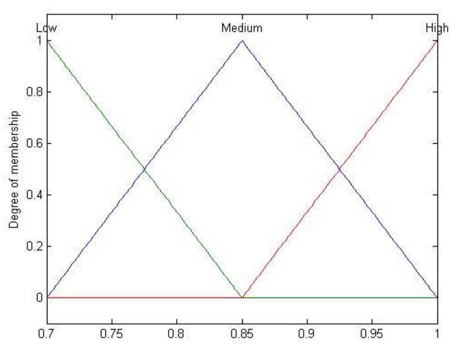


Figure 23. Output variable Pa of the FCS(Pa and Alpha)

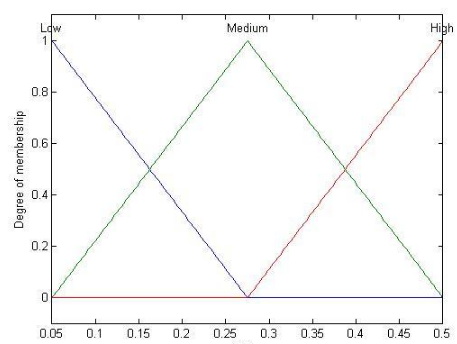


Figure 24. Output variable Alpha of the FCS(Pa and Alpha)

1. If (Iteration is Low) then (Pa is High)(Alpha is High) (1)
2. If (Iteration is Medium) then (Pa is Medium)(Alpha is Medium) (1)
3. If (Iteration is High) then (Pa is Low)(Alpha is Low) (1)

Figure 25. Rules of operation for the FCS(Pa and Alpha)

In this case we are using a fuzzy system called FCS (Pa and Alpha), and the structure of this fuzzy system is a follows: In Figure 22 we present the input variable “Iteration” and two outputs shown in Figures 23 and 24, the parameters are adjusted dynamically with type of triangular membership functions, with 3 rules shown in Figure 25.

7.6 PROPOSAL FOR THE FCS (ALPHA AND BETA)

The proposal for the variants of FCS (Alpha and Beta) is presented in Figure 26 and it consists of the dynamic setting of both parameters Alpha and Beta.

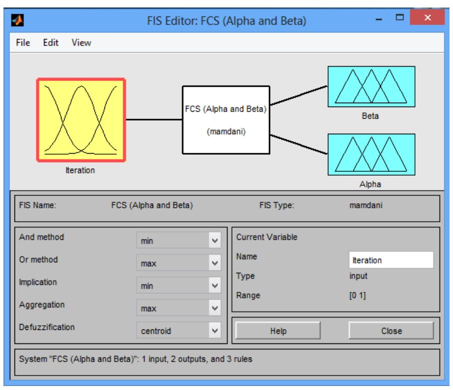


Figure 26. Proposal of FCS (Alpha and Beta)

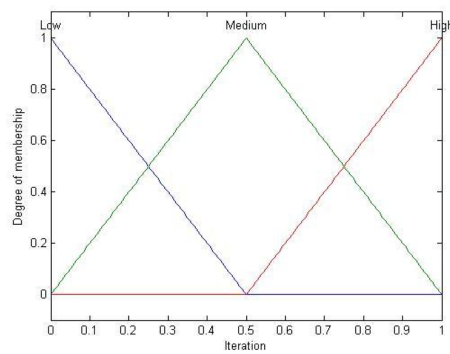


Figure 27. Input variable of FCS(Alpha and Beta)

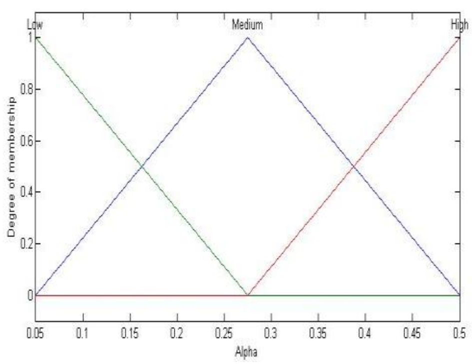


Figure 28. Output variable Alpha of FCS(Alpha and Beta)

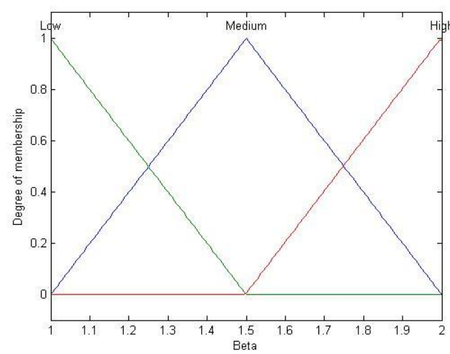


Figure 29. Output variable Beta of FCS(Alpha and Beta)

1. If (Iteration is Low) then (Beta is High)(Alpha is High) (1)
2. If (Iteration is Medium) then (Beta is Medium)(Alpha is Medium) (1)
3. If (Iteration is High) then (Beta is Low)(Alpha is Low) (1)

Figure 30. Rules of operation for the FCS(Alpha and Beta)

We consider as input to the fuzzy system the "Iteration" variable and two output variables, which are the alpha and beta parameters with a set of 3 fuzzy rules, which can be found in Figures 27 to 30.

7.7 PROPOSAL FOR FCS (PA, ALPHA AND BETA)

The main function of the fuzzy system called ‘FCS (Pa, Alpha and Beta)’ is to integrate all the parameters of the CS algorithm for their dynamic adjustment by implementing a fuzzy system to perform the simulations and compare to the other proposed variants. In particular, a comparison with the original method and another variant to determine is made to find out which one performs the best. In Figure 31 we present the proposal for FCS (Pa, Alpha and Beta):

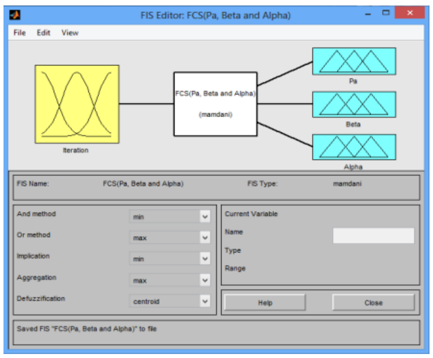


Figure 31. Poposal FCS (Pa, Alpha and Beta)

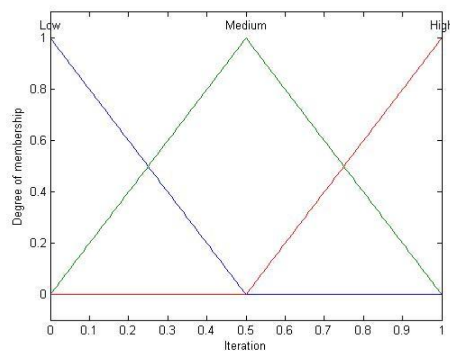


Figure 32. Input variable of FCS(Pa, Alpha and Beta)

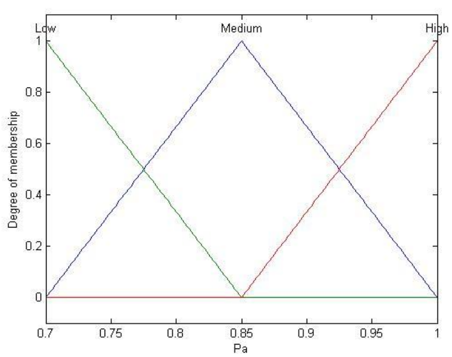


Figure 33. Output variable Pa of FCS(Pa, Alpha and Beta)

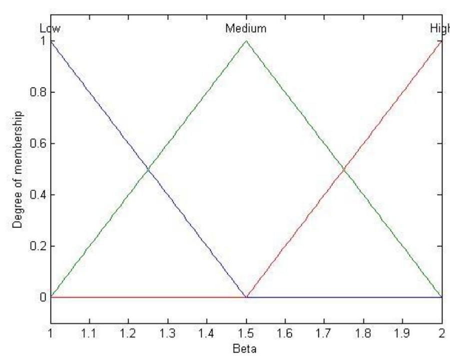


Figure 34. Output variable Beta of FCS(Pa, Alpha and Beta)

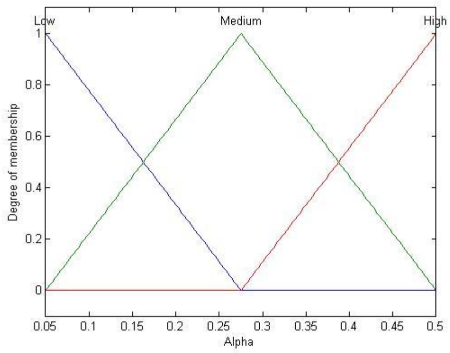


Figure 35. Output variable Alpha of FCS(Pa, Alpha and Beta)

1. If (Iteration is Low) then (Pa is High)(Beta is High)(Alpha is High) (1)
2. If (Iteration is Medium) then (Pa is Medium)(Beta is Medium)(Alpha is Medium) (
3. If (Iteration is High) then (Pa is Low)(Beta is Low)(Alpha is Low) (1)

Figure 36. Rules of operation for FCS(Pa, Alpha and Beta)

In this case, adjusting the parameters: Pa, Alpha and Beta, with the structure of 1 input, is presented in Figure 32, and 3 Outputs that are shown in Figures 33 to 35. The type of membership function is of Triangular form.

In Figure 36 we are presenting the rules of operation for this proposal, for example: "*If **Iteration** is **Low** the **Pa** is **High** and **Beta** is **High** and **Alpha** is **High***".

8 SIMULATION RESULTS

In this Section we discuss the parameters used for the simulations, the results of the comparisons with respect to the original method (Zhao & Li, 2012) a variant of the OCS algorithm (Zhao & Li, 2012) and our proposed variants, to determine which of the proposed FCS variants is the one that better helps convergence.

8.1 PARAMETERS USED IN THE SIMULATIONS

The ranges used for the above benchmark functions considered in the experiments are listed in Table 2 [30].

Table 2. Search space for each benchmark function

Function	Search space
F1	$(-5 \le x_i \le 5)$
F2	$(-5.12 \le x_i \le 5.12)$
F3	$(-600 \le x_i \le 600)$
F4	$(-15 \le x_i \le 30)$

Table 3. Parameters for each method

Parameters	**CS**	**OCS**	**FCS (Pa)**	**FCS (Beta)**	**FCS (Alpha)**	**FCS (Pa and Alpha)**	**FCS (Pa and Beta)**	**FCS (Pa, Alpha and Beta)**
Nest	25	25	25	25	25	25	25	25
Pa	75%	75%	Dynamic	75%	75%	Dynamic	Dynamic	Dynamic
β	1.5	1.5	1.5	Dynamic	1.5	1.5	Dynamic	Dynamic
α	5%	5%	5%	5%	Dynamic	Dynamic	5%	Dynamic
CR	NA	80%	NA	NA	NA	NA	NA	NA
D	15	15	15	15	15	15	15	15

*NA: Does Not Apply

In Table 3 we are presenting the parameters implemented in the simulations for each method, then a comparative set of experiments was performed with the original algorithm and the integration of fuzzy inference systems is presented, and the also compared with simulations implemented in Zhao [30].

In Table 4 we are presenting a summary of the experiments with the benchmark mathematical functions with all the proposed FCS variants and the original CS and the OCS approach presented in [30]. In Table 4 we are indicating in bold the best average results for the corresponding functions and we are also presenting other statistics of the experiments.

Table 4. Simulation results for each method

Function		**ALGORITHMS**							
				OUR PROPOSALS					
		CS	**OCS**	**FCS (Pa)**	**FCS (Beta)**	**FCS (Alpha)**	**FCS (Pa and Alpha)**	**FCS (Pa and Beta)**	**FCS (Pa, Alpha and Beta)**
F 1	Worst	15.9419	1.61E-04	4.44E-15	3.05E-33	7.86E-53	6.56E-53	1.50E-32	2.88E-20
	Average	12.6119	1.78E-05	4.44E-15	5.67E-34	8.47E-54	**8.38E-54**	1.54E-33	5.01E-21
	Best	7.445	1.96E-	4.44E-	2.52E-35	3.52E-55	5.69E-	5.30E-	4.41E-

		35	10	15			56	35	22
	S.D,	2.29873	3.90E-05	0	7.83E-34	1.51E-53	1.34E-53	2.78E-33	6.43E-21
F 2	Worst	129.512	8.209643	12.8122	3.3296185	0.994959	12.526557	13.656853	2.38E+01
	Average	112.2	0.541959	7.10837	1.0618771	**0.116149**	4.7098764	7.9871256	1.27E+01
	Best	92.3261	3.56E-07	2.82109	0.026066	1.48E-10	0.0001031	4.0281549	5.53E+00
	S. D.	10.0466	1.550602	2.25426	0.7617607	0.227964	2.8783038	2.2938975	4.01E+00
F 3	Worst	64.3236	0.42496	0.01256	0.0098721	0.007396	0.001169	0.0098578	0.060455
	Average	44.1934	0.052351	0.00075	0.0006613	0.001002	**3.90E-05**	0.001106	0.008032
	Best	18.5811	1.01E-09	0	1.12E-13	0	0	0	4.65E-14
	S. D.	11.6741	0.103029	0.00287	0.0025036	0.002508	0.0002134	0.0027902	0.012809
F 4	Worst	17.9939	0.084912	2.22E-14	1.96E-08	4.44E-15	4.44E-15	1.3404213	9.31E-01
	Average	16.1787	0.023539	6.10E-15	1.55E-09	**4.44E-15**	**4.44E-15**	0.0757242	9.31E-02
	Best	14.0585	3.48E-04	4.44E-15	3.40E-12	4.44E-15	4.44E-15	4.44E-15	9.31E-01
	S. D.	1.12286	0.023689	4.25E-15	4.06E-09	0	0	0.2931419	2.84E-01
Average for Algorithm		46.296	0.154467	1.77728	0.2656346	**0.029288**	1.18E+00	2.02E+00	3.20E+00

* S.D.: Standard Deviation

Table 4 presents the averages for all the algorithms, emphasizing in bold the best algorithm in each case, with an average of 0.029288 FCS (Alpha) is the best algorithm on average, which is better than CS with 46.2956 and the average 0.154467 [30].

We should also mention that all the algorithms that we have proposed are better than the original CS algorithm. In comparing with the F2 function, there was one variant of the proposed methods that did not beat the OCS method. However, the one who manages to overcome on average in all functions is FCS (Alpha) variants.

9 STATISTICAL TESTS

To perform the statistical tests we consider the following algorithms:

- CS original algorithm.
- OCS variant of the original algorithm with crossover proposed by Zhao.
- FCS (Alpha) proposed algorithm with dynamic adjustment in the alpha parameter.

In Table 4 we obtained the mean and standard deviations for each function and for each algorithm that we have considered in this statistical test. We consider the FCS (Alpha) variants of the algorithm because on average is the best algorithm to compare.
The statistical tests for comparison of the algorithms consider the z-test, and the parameters are defined in Table 5.

Table 5. Parameters for the statistical z- test

Parameter	Value
Level of significance	95%
Alpha	5%
Ho	$\mu_1 \geq \mu_2$
Ha	$\mu_1 < \mu_2$
Critical value	-1.645

Equation 13 shows the statistical "Z Test" that is used:

$$Z = \frac{\left(\overline{x_1} - \overline{x_2}\right) - \left(\overline{\mu_1} - \overline{\mu_2}\right)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{13}$$

The test results of the z-test statistics for CS [30] and the FCS (Alpha) algorithm for the 4 mathematical functions are shown in Table 6.

Table 6. Results of the z-test statistic for CS [30] and the FCS (Alpha) variant

Function	Value of Z	Evidence
F1	-30.0505	Significant
F2	-61.0902	Significant
F3	-20.7340	Significant
F4	-78.9187	Significant

The results of the z-test statistics for OCS [30] and the FCS (Alpha) algorithms for the 4 mathematical functions are shown in Table 7.

Table 7. Results of the test statistic z-test for OCS [30] and the FCS (Alpha) variant

Function	Valor de Z	Evidence
F1	-2.4999	Significant
F2	-1.4881	Not Significant
F3	-2.7298	Significant
F4	-5.4425	Significant

When performing the statistical tests "z-test" by function, with a significance level of 5%, the Ha (alternative hypothesis) indicates that the average of the proposed method is less than the average of the original method CS. In addition, there is sufficient reason to reject the Ho (null hypothesis) with a rejection region for all values lower than -1.645, and therefore the algorithm FCS (Alpha) has sufficient statistical evidence to outperform the original CS algorithm.

For the statistical test of the functions for the FCS and OCS algorithms there is insufficient statistical evidence in the F2 function, but for the other functions there is enough evidence to reject Ho (null hypothesis). It is noteworthy that despite having insufficient evidence in the F2 function, the rejection region is minimal. But on average there is enough evidence to say that the FCS (Alpha) algorithm presents in most of the functions sufficient evidence to be considered the best algorithm.

10. CONCLUSIONS

We have performed the design of experiments with different ranges of the parameters to evaluate the performance of the CS algorithm and propose the implementation of the fuzzy logic systems for dynamic adjustment of parameters during execution and obtain the best algorithm according to the statistical tests to prove their efficiency.

In recent times fuzzy logic has allow to solve problems as human beings would resolve through decision rules, and this idea was implemented in our proposals to help convergence. With this proposed approach we have obtained good results.

Computational results and detailed comparison studies with CS [30], OCS show that the FCS algorithm presents promising results when compared, to the variants and even the original algorithm.

The results of the simulations and comparisons of our proposal with the original CS and OCS [30], indicate that on average our proposal is better than the original algorithm and the OCS variant. A comparison between our fuzzy approaches for improving CS produces that the best results are presented by the FCS (Alpha) variant because the Alpha parameter accelerates convergence. Statistics shows that there is sufficient evidence to accept the alternative hypothesis and reject the null hypothesis, which states that our proposed algorithms have better results when compared to the original CS algorithm. Also the proposed FCS (Alpha) variant of the algorithm has better results when compared with the OCS algorithm [30].

REFERENCES

[1] Belarbi, K., Titel, F., Bourebia, W., & Benmahammed, K., Design of Mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm. Engineering applications of artificial intelligence, 18(7), 875-880. , 2005

[2] Bonabeau, E., Dorigo, M., & Theraulaz, G.. Swarm Intelligence: From Natural to Artificial Systems. No. 1 Oxford University Press., 1999

[3] Das, S., Dasgupta, P., & Panigrahi, B. K., Inter-species Cuckoo Search via Different Levy Flights. In Swarm, Evolutionary, and Memetic Computing, Springer International Publishing, 515-526, 2013

[4] Ding, X., Xu, Z., Cheung, N. J., & Liu, X., Parameter estimation of Takagi–Sugeno fuzzy system using heterogeneous cuckoo search algorithm. Neurocomputing, 151, 1332-1342., 2015.

[5] Edwards, A. M., Phillips, R. A., Watkins, N. W., Freeman, M. P., Murphy, E. J., Afanasyev, V, & Viswanathan, G. M., Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. Nature, 449(7165), 1044-1048, 2007

[6] Eiben, A.E., Smith, J.E., Introduction to evolutionary computing (natural computing series). Springer, Berlin, 2013

[7] Ghanbari, A., & Moghanni-Bavil-Olyaei, M. R., Adaptive fuzzy terminal sliding-mode control of MEMS z-axis gyroscope with extended Kalman filter observer. Systems Science & Control Engineering: An Open Access Journal, 2(1), 183-191, 2014.

[8] Ghodrati, A., Lotfi, S., A hybrid CS/GA algorithm for global optimization. In: Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011), 20–22 December 2011, Springer India, 397–404, 2012.

[9] Ghodrati, A., Lotfi, S., A hybrid CS/PSO algorithm for global optimization. In: Intelligent Information and Database Systems, Springer Berlin Heidelberg, 89–98, 2012.

[10] Guerrero M., Castillo O., García M., Cuckoo Search via Lévy Flights and a Comparison with Genetic Algorithms, Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics, Studies in Computational Intelligence, Springer, Volume 574, 91-103, 2015. DOI: 10.1007/978-3-319-10960-2_6

[11] Jamil, M., Yang, X.-S., A literature survey of benchmark functions for global optimization problems. Int. J. Math. Model. Numer. Optim. 4(2), 150–194, 2013.

[12] Jamil, M., Zepernick, H., Multimodal function optimization with cuckoo search algorithm. Int. J. Bio-Inspired Comput. 5(2), 73–83, 2013.

[13] Layeb A., A novel quantum-inspired cuckoo search for Knapsack problems. Int J Bioinspired Comput 3(5), 297–305, 2011.

[14] Li, X., Yin, M., A hybrid cuckoo search via lévy flights for the permutation flow shop scheduling problem. Int. J. Prod. Res. (ahead-of-print), 1–23, 2013.

[15] Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., & Valdez, M., Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. Expert Systems with Applications, 40(8), 3196-3206, 2013.

[16] Ong, P., Zainuddin, Z., An efficient cuckoo search algorithm for numerical function optimization. In: AIP Conference Proceedings, vol. 1522, 1378, 2013.

[17] Pandey, A., & Parhi, D. R., MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule based Fuzzy Logic Controller. Procedia Technology, 14, 28-34, 2014.

[18] Pavlyukevich, I., Cooling down Lévy flights. J. Phys. A. Math. Theor. 40, 12299–12313, 2007.

[19] Plummer, E. A., Time series forecasting with feed-forward neural Networks: guidelines and limitations. University of Wyoming, July, 2000.

[20] Pulido, M., & Melin, P. , Ensemble Neural Network Optimization Using the Particle Swarm Algorithm with Type-1 and Type-2 Fuzzy Integration for Time Series Prediction. In Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer International Publishing, 99-112, 2014.

[21] Reynolds, A. M., & Frye, M. A. , Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search. PLoS One Vol. 2, Num. 4, e354, 2007.
[22] Yang, X. S. Nature-inspired metaheuristic algorithms. Luniver press.
[23] Yang, X.-S., Deb, S. Cuckoo search via Lévy flights, in Proceedings of The World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE, Conference 9-11, December, 210-214.
[24] Yang, X.-S., Deb, S., Engineering optimization by cuckoo search. Int. J.Math. Model. Numer. Optim. 1(4), 330–343, 2010.
[25] Ye, Z., Wang, M., Hu, Z., & Liu, W., An Adaptive Image Enhancement Technique by Combining Cuckoo Search and Particle Swarm Optimization Algorithm. Computational Intelligence and Neuroscience, 2015.
[26] Yildiz AR., Cuckoo search algorithm for the selection of optimal machine parameters in milling operations. Int J Adv Manuf Technol,, 2012. DOI:10.1007/s00170-012-4013-7
[27] Zadeh, L.,. Fuzzy sets. Information & Control, 8., 1965.
[28] Zadeh, L., Fuzzy logic. IEEE Computer, pp. 83–92, 1965.
[29] Zadeh, L., Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems, 90, 177, 1997.
[30] Zhao, P., & Li, H., Opposition-based Cuckoo search algorithm for optimization problems. In Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on IEEE , Vol. 1, 344-347, 2012.
[31] Zhou Z.-H., Jiang Y., Yang Y.-B., and Chen S.-F., Lung cancer cell identification based on artificial neural network ensembles, Artificial Intelligence in Medicine 24 (1), 25-36, 2002.

TIJUANA INSTITUTE OF TECHNOLOGY, TIJUANA MÉXICO
Current address: Tijuana Institute of Technology, Tijuana México.
E-mail address: ocastillo@tectijuana.mx