algorithms

MDPI

Article

Imperialist Competitive Algorithm with Dynamic Parameter Adaptation Using Fuzzy Logic Applied to the Optimization of Mathematical Functions

Emer Bernal, Oscar Castillo *, José Soria and Fevrier Valdez

Tijuana Institute of Technology, Tijuana 22379, Mexico; emer_recargado@hotmail.com (E.B.); jsoria57@gmail.com (J.S.); fevrier@tectijuana.mx (F.V.)

* Correspondence: ocastillo@tectijuana.mx; Tel.: +52-664-623-6318

Academic Editors: Yun-Chia Liang and Toly Chen

Received: 28 September 2016; Accepted: 16 January 2017; Published: 23 January 2017

Abstract: In this paper we are presenting a method using fuzzy logic for dynamic parameter adaptation in the imperialist competitive algorithm, which is usually known by its acronym ICA. The ICA algorithm was initially studied in its original form to find out how it works and what parameters have more effect upon its results. Based on this study, several designs of fuzzy systems for dynamic adjustment of the ICA parameters are proposed. The experiments were performed on the basis of solving complex optimization problems, particularly applied to benchmark mathematical functions. A comparison of the original imperialist competitive algorithm and our proposed fuzzy imperialist competitive algorithm was performed. In addition, the fuzzy ICA was compared with another metaheuristic using a statistical test to measure the advantage of the proposed fuzzy approach for dynamic parameter adaptation.

Keywords: imperialist competitive algorithm; fuzzy logic; dynamic adjust

1. Introduction

Swarm intelligence techniques have gained popularity in recent decades because of their capacity to locate partially optimal solutions for combinatorial optimization problems. These have been applied in various areas, such as engineering, economics and industry, etc., and these problems benefit from the use of swarm intelligence techniques because they are usually very hard to solve accurately since there is no precise algorithm to solve them [1,2].

Swarm intelligence techniques are approximate metaheuristics that include a wide range of smart algorithms normally inspired in natural processes, such as artificial honey bee (AHB), genetic algorithm (GA), cuckoo search (CS), and gravitational search algorithm (GSA) [3].

The imperialist competitive algorithm (ICA), was proposed in 2007 by Atashpaz-Gargari and Lucas. ICA was originated on the idea of imperialism and in this process stronger countries try to colonize the weakest countries and make them part of their colonies [4]. This algorithm has been currently used in different industrial applications [5].

ICA was initially used in continuous optimization problems, but now it has been utilized on many complex optimization problems such as flowlines scheduling problems (FSP), traveling salesman problem (TSP), assembly line balancing problem (ALBP), and facility line design problem (FLP) [1].

In the recent literature, there are some articles where the imperialist competitive algorithm has been studied, such as the work on an imperialist competitive algorithm to optimize artificial neural networks for UCAV global path planning [6], where the competitive imperialist algorithm is used to train a neural network with which it is possible to reduce uncertainty and avoid falling into local minimum. Another paper is that of a hybrid imperialist competitive algorithm for minimizing make

span in a multi-processor open shop [1]. In this case, a new linear programming model is proposed for programming problems of a multiprocessor store to minimize the make span, in which a hybrid imperialist competitive algorithm (ICA) with the genetic algorithm (GA) is used. Another work is the application of an imperialist competitive algorithm to the design of a linear induction motor [7]. In this paper, the competitive imperialist algorithm is used to design a low speed linear induction motor to improve efficiency with a high power factor. Finally, there is also a paper on the Imperialist Competitive Algorithm for Solving a Dynamic Cell Formation Problem with Production Planning [8]. This article uses the competitive imperialist algorithm to optimize the planning, production and reconfiguration costs in the cell formation with production planning.

The imperialist competitive algorithm has been used in different applications and in different branches of engineering, but there are some aspects that can be explored to improve its performance, like the hybridization of the imperialist competitive algorithm with different metaheuristics, such as GA, PSO and other population based metaheuristics [3,9,10]. Since we can use the ICA algorithm as a starting point to create an initial solution or use the algorithm as a tool to promote exploration or exploitation. One aspect to be taken into account is to use the competitive imperialist algorithm in parallel to help accelerate the search and to achieve higher quality in the possible solutions. Another important alternative is to perform the adjustment of the ICA algorithm parameters to improve the algorithm efficiency as is the idea presented in this research.

After analyzing the existing literature, it has been observed that until now there are no works where the dynamic adjustment of the parameters has been done using fuzzy logic and very few where the imperialist competitive algorithm for the minimization of mathematical functions has been used. We conclude that realizing the dynamic adjustment of the parameters in the imperialist competitive algorithm utilizing fuzzy logic to carry out the adjustment of the parameters that influences the operation of the algorithm, which could help to improve performance and provide tools to address the uncertainty generated during operation of the algorithm.

The study of the metaheuristic is realized to observe the efficiency of the imperialist competitive algorithm (ICA) when used in optimization problems [11,12], having the original ICA algorithm as a basis for modifying the algorithm in dynamically performing adaptation of parameters. In this case, this has been proven to be a good idea in other metaheuristics, which utilize adaptation of parameters along the iterations to help improve the results obtained with respect to when static or fixed parameters are utilized.

To realize this modification utilizing fuzzy logic [13,14], we continued to the application of the algorithm for benchmark mathematical functions [10]. In this case the idea is to observe the results of the ICA algorithm and it is expected to obtain good or better results than with the original algorithm and other metaheuristics. In this regard, statistical tests were conducted to compare the results of the original algorithm, the modified fuzzy ICA and other metaheuristics.

The article is organized as shown below: in Section 2 the methodology of the original Imperialist Competitive algorithm (ICA) is explained, that is, the equations used for assimilation, the power of empires and each of the algorithm parameters, an explanation is also given on how the parameters can be changed throughout the iterations. Section 3 presents the methodology that was followed for modifying the ICA algorithm using fuzzy logic. In Section 4 the results obtained with the ICA algorithm and the fuzzy imperialist competitive algorithm (FICA) are presented. In Section 5 the results of the comparison between fuzzy imperialist competitive algorithm (FICA) and the original ICA algorithm are presented. In Section 6 we show the statistical test performed between the fuzzy imperialist competitive algorithm (FICA) and fuzzy Cuckoo Search (FCS) and finally Section 7 describes the conclusions.

2. Imperialist Competitive Algorithm

This section provides a description of the original imperialist competitive algorithm that was proposed by Atashpaz-Gargari and Lucas, the corresponding equations, and the flowchart used to

understand its operation [15]. In the field of metaheuristics, ICA is based on political and human social progress, unlike other evolutionary metaheuristic algorithms, based on behaviors of animals or physical phenomena [16,17].

Figure 1 shows the flowchart of the imperialist competitive algorithm. Like other evolutionary or population based algorithms, ICA starts with an initial population. The best countries are selected as imperialist countries and the rest form the imperialist colonies. All colonies of the initial population are divided among the imperialists countries based on of their power [18].

After dividing all the colonies among the imperialists, the colonies begin the movement towards their imperialist countries. The total power of all the empires depend on the power of the imperialist countries and the power of their colonies. This fact is defined with the total power of an empire by the power of the imperialist country plus a percentage of the average power of their colonies [15].

When the imperialist competition between the empires starts, any empire that cannot remain in this competition and cannot augment its power will be removed from the imperialist competition. The imperialist competition will result in an increment in the power of the most powerful empires and a decrement in the power of the weaker empires. The weakest empires will lose their power partially and, ultimately, will collapse one by one. All countries will eventually become a state in which there is only one empire in the entire world and all remaining countries are colonies of that empire [19,20].

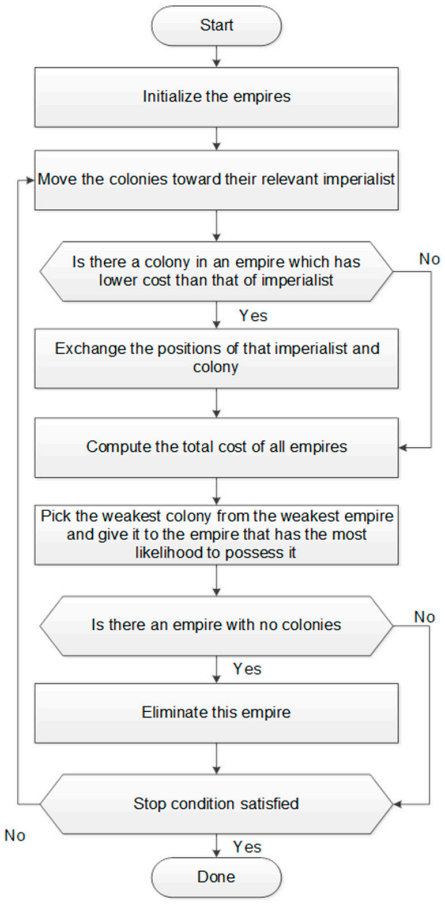

Figure 1. Imperialist competitive algorithm flowchart.

The equations for the formation of empires (Initialization) are the following [21]:

$$Country = [p_1, p_2, \ldots, p_{Nvar}] \tag{1}$$

$$Cost = f\ (Country) = f\ (p_1, p_2, \ldots, p_n) \tag{2}$$

$$N_{coI} = N_{pop} - N_{imp} \tag{3}$$

$$C_n = \max_i\{c_i\} - c_n \tag{4}$$

$$p_n = \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \tag{5}$$

$$NC_n = \text{Round}\{p_n N_{col}\} \tag{6}$$

In Equation (1) *Country* represents a country, *Nvar* is the number of variables of interest and p_i is the value of i-th variable, Equation (2) represents the cost of a country, Equation (3) is necessary to obtain the number of colonies of the population, Equation (4) is used to obtain the normalized cost of each imperialist, where c_n is the n-th imperialist's cost, and C_n is the normalized cost of n-th imperialist, Equation (5) obtains the power of each imperialist where p_n is the power of n-th imperialist and Equation (6) represents the number of colonies that can possess that imperialist.

The equations for moving the colonies towards their imperialist country (assimilation) are [15]:

$$x \sim U(0, \beta d) \tag{7}$$

$$\theta \sim U(-\gamma, \gamma) \tag{8}$$

Equation (7) represents a colony that moves a distance x towards its imperialist, where β is a number between 1 and 2 and d represents the distance among the colony and its imperialist, Equation (8) helps us to search for different positions around the imperialist. In which γ is a parameter where a large value estimates a global search and a small value impacts the local search.

The Equation for the full power of an empire is as follows [21,22]:

$$TC_n = Cost(imp) + \xi \text{mean}\{Cost(Col)\} \tag{9}$$

This equation represents the total cost of n-th empire and ξ is a number between 0 and 1. Where with a small value of ξ, we have a greater influence of the imperialist power to determine the total power of the empire, and a large value of ξ, has a greater influence on the average power of the colonies to calculate the total power of the empire.

The equations for the imperialist competition are the following [21]:

$$NTC_n = \max_i\{TC_i\} - TC_n \tag{10}$$

$$p_{p_n} = \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \text{ where } \sum_{i=1}^{N_{imp}} p_{p_i} = 1 \tag{11}$$

Equation (10) starts with the imperialist competition where the normalized total cost is calculated and in Equation (11) we provide the probability of possessing a colony. The main steps of the algorithm are shown below in the pseudocode [23,24].

2.1. Pseudocode ICA

1. Initialize the empires (Equations (1)–(6)).

   $Country = [p_1, p_2, ..., p_{Nvar}]$
   $Cost = f\ (Country) = f\ (p_1, p_2, ..., p_n)$
   $N_{coI} = N_{pop} - N_{imp}$
   $C_n = \max_i\{c_i\} - c_n$
   $p_n = \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}$
   $NC_n = \text{Round}\{p_n N_{col}\}$

2. Move all colonies toward their imperialist country (Assimilating Equations (7) and (8)).

   $x \sim U(0, \beta d)$
   $\theta \sim U(-\gamma, \gamma)$

3. If there is a colony with has less cost than the imperialist country in an empire, exchange the locations of the colony and the imperialist country.
4. Calculate total cost of each empire (Equation (9)).

   $TC_n = Cost(imp) + \xi \text{mean}\{Cost(Col)\}$

5. Select the weaker colony from the weaker empire and pass it to the empire that has more possibility to possess it. (The imperialistic competition Equations (10) and (11)).

   $NTC_n = \max_i\{TC_i\} - TC_n$
   $p_{p_n} = \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i}$

6. Delete the weakest empires.
7. If only one empire remains, stop, if not return to step 2.

2.2. Mathematical Functions

In this section, the mathematical functions utilized in the tests to measure the performance of the ICA algorithm for the dynamic adjustment of its parameters are listed below. In the field of metaheuristics used for optimization problems it is usual to consider mathematical functions to measure their performance, as it is used in this paper. In this case, the idea is to measure an improvement of an optimization algorithm known as ICA, where the aim is to use dynamic parameters [13,25]:

Figure 2 shows the plot of the sphere function, Figure 3 shows the quartic function, Figure 4 illustrates the plot of the Rosenbrock function, Figure 5 shows the Rastrigin function, and the Griewank function is presented in Figure 6 and finally in Figure 7 shows the plot of the Ackley function. All the mathematical functions that are used are accompanied by their respective equation and the search space in which they work as shown below [16,26].

- Sphere

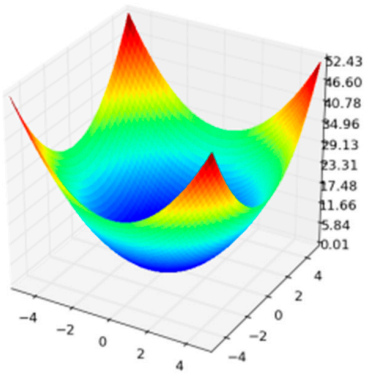

Figure 2. Sphere Function.

$$f(x) = \sum_{j=1}^{n_x} x_j^2 \tag{12}$$

Search space $x_j \in [-5.12, 5.12]$ and $f^*(x) = 0.0$

- Quartic

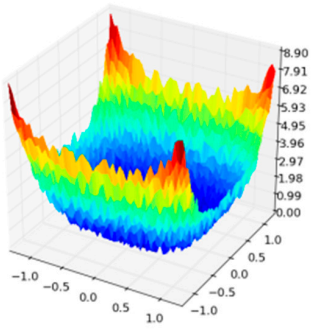

Figure 3. Quartic Function.

$$f(x) = \sum_{i=1}^{n} i x_i^4 \tag{13}$$

Search space $x_i \in [-1.28, 1.28]$ and $f^*(x) = 0.0$

- Rosenbrock

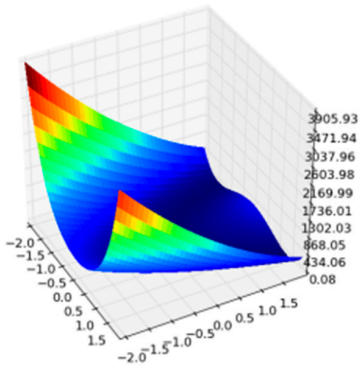

Figure 4. Rosenbrock Function.

$$f(x) = \sum_{j=1}^{n_z/2} [100(x_{2j} - x_{2j-1}^2)^2 + (1 - x_{2j-1})^2] \tag{14}$$

Search space $x_j \in [-2.048, 2.048]$ and $f^*(x) = 0.0$

- Rastrigin

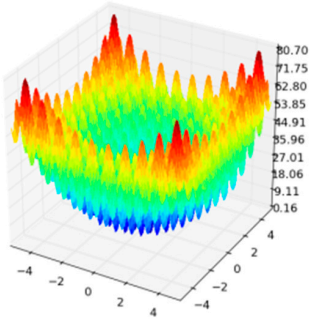

Figure 5. Rastrigin Function.

$$f(x) = \sum_{j=1}^{n_x} (x_j^2 - 10\cos(2\pi x_j) + 10) \tag{15}$$

Search space $x_j \in [-5.12, 5.12]$ and $f^*(x) = 0.0$

- Griewank

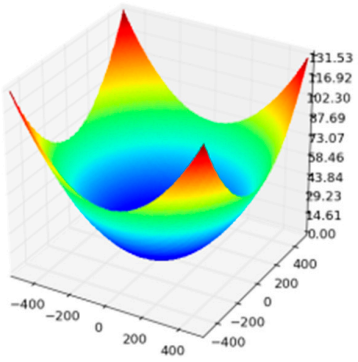

Figure 6. Griewank Function.

$$f(x) = 1 + \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) \tag{16}$$

Search space $x_i \in [-512, 512]$ and $f^*(x) = 0.0$

- Ackley

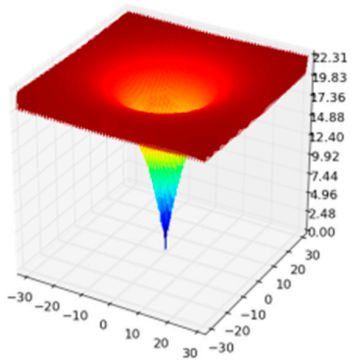

Figure 7. Ackley Function.

$$f(x) = -20e^{-0.2\sqrt{\frac{1}{n_x}\sum_{j=1}^{n_x} x_j^2} - \frac{1}{e^{n_x}}\sum_{j=1}^{n_x}\cos(2\pi x_j)} + 20 + e \tag{17}$$

Search space $x_j \in [-30, 30]$ and $f^*(x) = 0.0$

3. Proposal Methodology

The Imperialist competitive algorithm is a search technique that takes as inspiration the theory of imperialism, where the most powerful countries intent to make a colony of other countries and currently been utilized to resolve complex optimization problems. A metaheuristic algorithm named Fuzzy imperialist competitive algorithm (FICA) with dynamic adjustment of parameters used on the optimization of benchmark mathematical functions is proposed in this article. The main idea is that

fuzzy logic will help in modeling the uncertainty in finding the appropriate parameter values during the execution of the algorithm and in this way improve performance of the algorithm.

The main objective of our proposal is to obtain the optimal values of the parameters with the help of fuzzy systems to increase performance of the ICA algorithm during execution. The fuzzy systems [13] will dynamically adjust the β and ξ parameters as shown below in Figure 8.

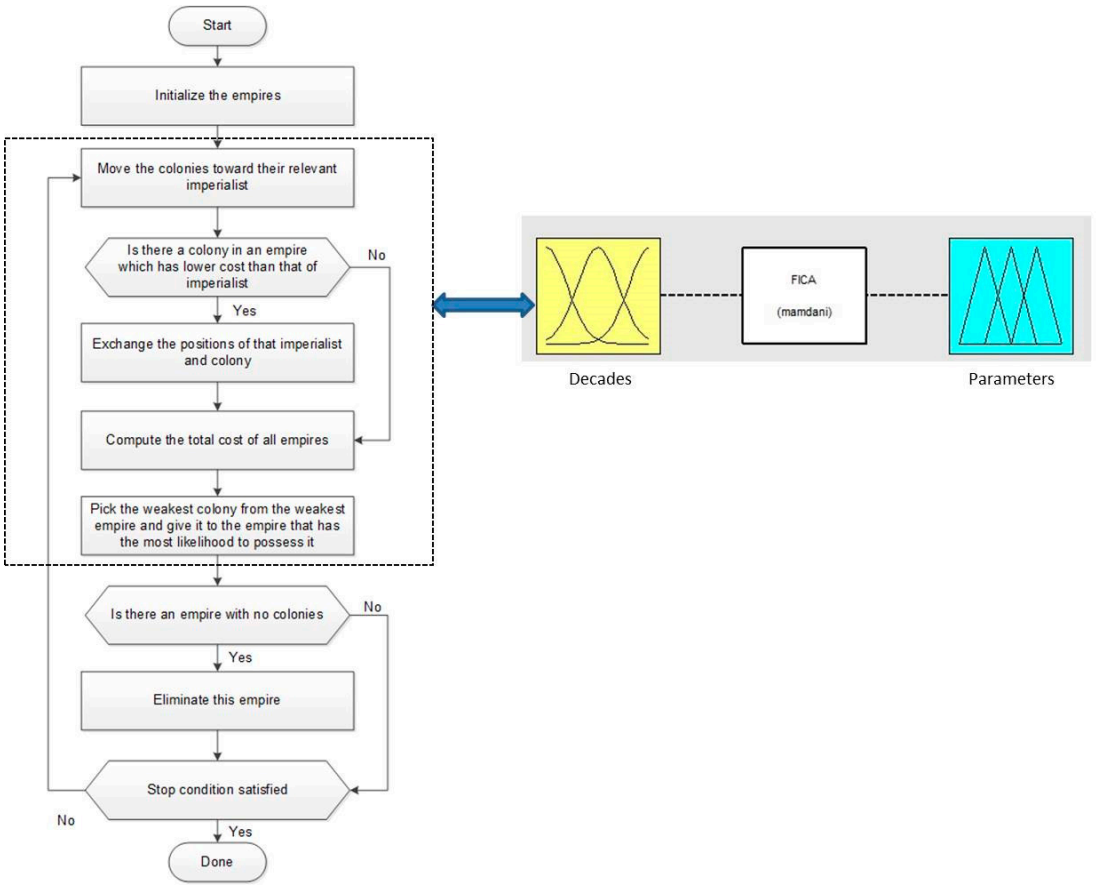

Figure 8. The proposed fuzzy imperialist competitive algorithm (FICA).

For the decades of the algorithm, we did perform an analysis and decided to utilize a percentage of decades with respect to the total number of decades, that is, when the algorithm starts the decades will be considered to have a linguistic value of "low", and when most of the decades have passed will be considered "high" or close to 100%. This notion is presented below [27]:

$$\text{Decades} = \frac{\text{Current Decade}}{\text{Total Number Of Decades}} \tag{18}$$

This paper proposes and explains three fuzzy systems where the experiments were conducted. We have various fuzzy systems for obtaining the β parameter, the ξ parameter and a fuzzy system for the combination between β and ξ parameters.

The key issue is how to correctly define the fuzzy systems. In this regard, the fuzzy systems shown in Figures 9–11 are fuzzy systems of Mamdani type with the input defined as the decades and with one output variable, the first is with the β parameter, the second with the ξ parameter and the third one with a combination of β and ξ.

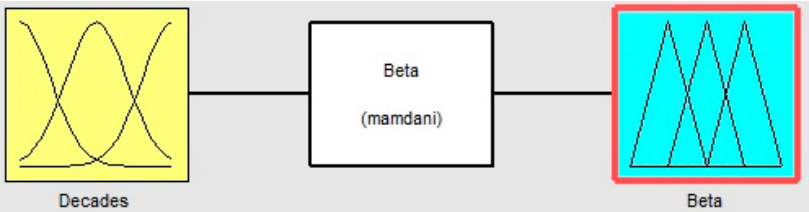

Figure 9. FICA β.

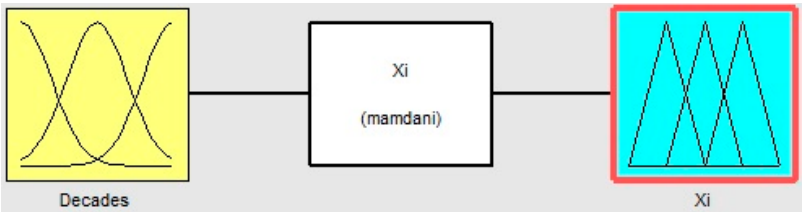

Figure 10. FICA ξ.

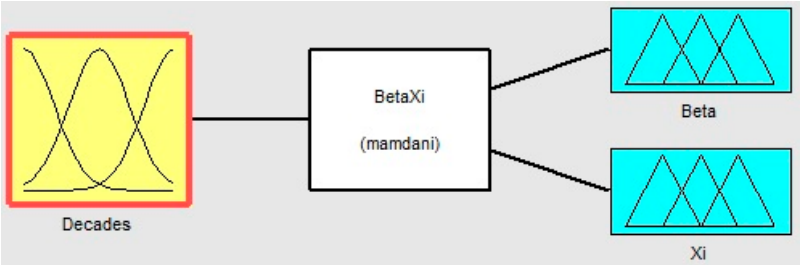

Figure 11. FICA combination β and ξ.

The design of the input variable is found in Figure 12, which represents the decades; this input variable is divided in three triangular membership functions tagged as Low, Medium and High.

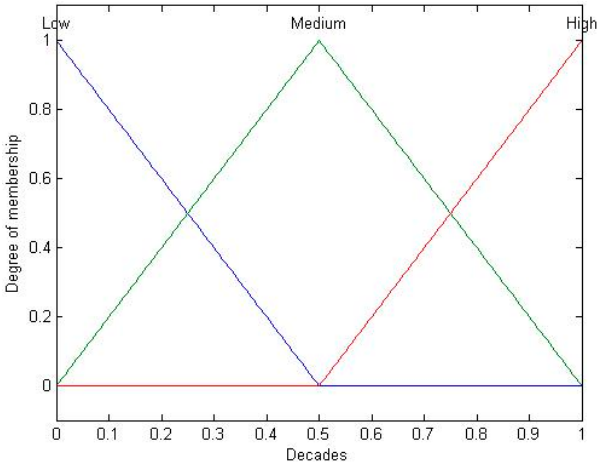

Figure 12. Input called Decades.

The Fuzzy systems have the output variables shown in Figures 13 and 14, and are β and ξ, respectively, and in the same way that the input variables are granulated into three triangular membership functions tagged as Low, Medium and High.

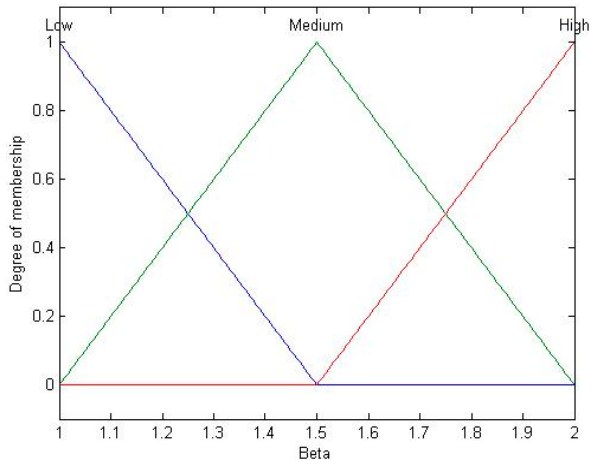

Figure 13. Output called β.

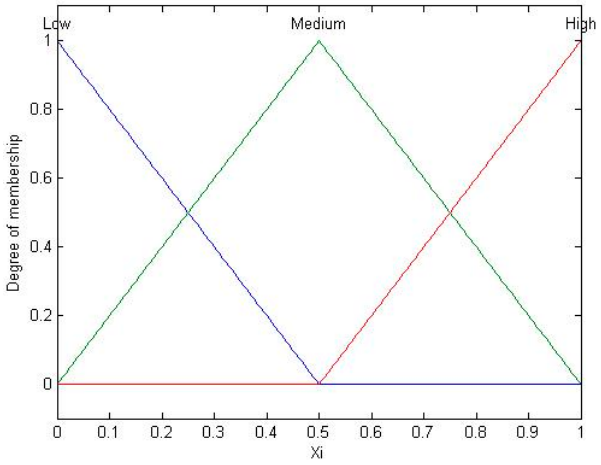

Figure 14. Output called ξ.

For the design of the rules of the fuzzy systems, it was determined that in the initial decades the ICA algorithm begins to explore and when approaching the end eventually exploits. All the rules were designed taking the idea that they are in increasing fashion as shown below:

The rules of the fuzzy system for β are:

1. If (Decades is Low) then (β is Low).
2. If (Decades is Middle) then (β is Medium).
3. If (Decades is Alto) then (β is High).

The rules of the fuzzy system for ξ are:

1. If (Decades is Low) then (ξ is Low).
2. If (Decades is Middle) then (ξ is Medium).
3. If (Decades is Alto) then (ξ is High).

To design the rules of the fuzzy system combining β and ξ, we decided (based on previous experimentation) to use β in increase over the decades and ξ that was in decrement as shown below:

1. If (Decades is Low) then (β is Low) and (ξ is High).
2. If (Decades is Medium) then (β is Medium) and (ξ is Medium).
3. If (Decades is High) then (β is High) and (ξ is Low).

4. Simulation Results with the Fuzzy Systems of the Imperialist Competitive Algorithm for Benchmark Mathematical Functions

In this section the imperialist competitive algorithm (ICA) [28,29] is tested with 6 mathematical functions with 30 dimensions for each of the β, ξ parameters and a combination of β and ξ and the results that were obtained by the ICA algorithm and our fuzzy ICA proposal are shown in separate tables for each function. All tables show the average, worst and best values obtained after 30 executions for algorithm from 1000 to 5000 decades.

The parameters utilized in the imperialist competitive algorithm [30] and the fuzzy imperialist competitive algorithms are:

- Dimensions: 30
- No. Countries: 200
- No. Imperialists: 10
- Revolution rate: 0.2

Table 1 shows the results of executing 30 times the proposed ICA algorithm with dynamic adaptation the β and ξ parameters, where we find the average, best and worst results obtained for the sphere function.

Table 1. Sphere function.

Sphere Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	3.11×10^{-31}	7.70×10^{-28}	4.59×10^{-31}	3.35×10^{-27}
	Worst	7.51×10^{-20}	1.71×10^{-24}	6.39×10^{-18}	7.98×10^{-22}
	Mean	2.51×10^{-21}	2.28×10^{-25}	2.50×10^{-19}	6.37×10^{-23}
2000	**Best**	1.68×10^{-61}	1.94×10^{-57}	5.05×10^{-62}	1.26×10^{-56}
	Worst	3.55×10^{-48}	9.63×10^{-51}	3.62×10^{-27}	3.91×10^{-49}
	Mean	1.76×10^{-49}	5.08×10^{-52}	1.21×10^{-28}	2.59×10^{50}
3000	**Best**	6.42×10^{-90}	3.66×10^{-87}	3.56×10^{-90}	2.14×10^{-85}
	Worst	1.33×10^{-52}	4.04×10^{-76}	7.56×10^{-25}	7.24×10^{-75}
	Mean	4.43×10^{-54}	1.53×10^{-77}	2.52×10^{-26}	2.86×10^{-76}
4000	**Best**	1.66×10^{-123}	3.27×10^{-110}	1.19×10^{-122}	3.05×10^{-110}
	Worst	1.67×10^{-71}	2.21×10^{-97}	1.15×10^{-47}	9.98×10^{-97}
	Mean	5.56×10^{-73}	9.51×10^{-99}	3.84×10^{-49}	3.34×10^{-98}
5000	**Best**	3.25×10^{-154}	7.20×10^{-138}	8.53×10^{-154}	7.05×10^{-133}
	Worst	4.50×10^{-121}	4.24×10^{-20}	9.90×10^{-16}	2.55×10^{-116}
	Mean	1.50×10^{-122}	1.42×10^{-21}	3.30×10^{-17}	1.68×10^{-117}

Table 2 shows the results of running 30 times the proposed ICA algorithm with dynamic adaptation of the β and ξ parameters, where we find the average, best and worst results obtained for the quartic function.

Table 2. Quartic function.

Quartic Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	4.03×10^{-55}	7.48×10^{-45}	1.85×10^{-55}	9.33×10^{-44}
	Worst	2.93×10^{-39}	8.69×10^{-38}	6.59×10^{-41}	1.28×10^{-35}
	Mean	9.76×10^{-41}	2.97×10^{-39}	2.39×10^{-42}	8.43×10^{-37}
2000	**Best**	2.34×10^{-113}	2.00×10^{-95}	4.70×10^{-108}	1.87×10^{-92}
	Worst	1.32×10^{-87}	5.40×10^{-85}	3.72×10^{-89}	8.94×10^{-82}
	Mean	4.39×10^{-89}	1.85×10^{-86}	1.26×10^{-90}	3.61×10^{-83}

Table 2. *Cont.*

Quartic Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
3000	**Best**	9.62×10^{-168}	1.77×10^{-141}	8.50×10^{-169}	2.74×10^{-140}
	Worst	5.57×10^{-147}	1.48×10^{-128}	4.20×10^{-148}	1.26×10^{-123}
	Mean	2.14×10^{-148}	4.98×10^{-130}	1.75×10^{-149}	4.24×10^{-125}
4000	**Best**	5.26×10^{-223}	8.31×10^{-190}	1.54×10^{-231}	3.06×10^{-181}
	Worst	1.35×10^{-195}	3.00×10^{-161}	6.16×10^{-196}	1.58×10^{-163}
	Mean	4.52×10^{-197}	1.00×10^{-162}	2.05×10^{-197}	5.59×10^{-165}
5000	**Best**	9.13×10^{-281}	2.93×10^{-234}	6.39×10^{-282}	1.16×10^{-237}
	Worst	1.51×10^{-240}	1.95×10^{-195}	2.50×10^{-245}	7.92×10^{-199}
	Mean	5.44×10^{-242}	6.52×10^{-197}	8.34×10^{-247}	2.71×10^{-200}

Table 3 shows the results of executing 30 times the proposed ICA algorithm with dynamic adaptation the β and ξ parameters, where we find the average, best and worst results obtained for the Rosenbrock function.

Table 3. Rosenbrock function.

Rosenbrock Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	1.2316226	2.4035099	2.8943449	1.741879
	Worst	20.229215	26.258041	26.428196	24.244927
	Mean	18.32843	17.302077	18.708605	17.302317
2000	**Best**	0.0182118	0.0019663	0.0309808	0.0212138
	Worst	19.389756	15.862588	19.380454	19.806802
	Mean	10.702829	9.026535	10.575362	11.571011
3000	**Best**	9.35×10^{-6}	0.0154982	2.24×10^{-6}	0.0907321
	Worst	13.231482	10.229455	13.023777	15.778142
	Mean	5.0096391	3.5266455	5.1161619	7.0405932
4000	**Best**	1.26E-07	7.56×10^{-5}	6.79×10^{-7}	0.0030788
	Worst	7.2442467	4.5619828	12.048154	5.1284706
	Mean	2.518922	0.8121844	2.5097263	1.8056573
5000	**Best**	8.87×10^{-7}	2.49×10^{-5}	7.28×10^{-9}	0.0089079
	Worst	9.102693	4.0085791	10.12155	4.1014126
	Mean	2.0320047	1.0782768	1.5795989	0.6206618

Table 4 summarizes the results of running 30 times the proposed ICA algorithm with dynamic adaptation the β and ξ parameters, where we find the average, best and worst results obtained for the Rastrigin function.

Table 4. Rastrigin function.

Rastrigin Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	77.60654371	33.89704822	101.485221	61.40540354
	Worst	166.3705786	157.4160653	174.25906	151.2330801
	Mean	131.0164965	95.8100497	125.297209	108.721887
2000	**Best**	66.66206493	3.542874325	46.762995	4.320732599
	Worst	156.421058	115.4147346	153.222524	117.6886745
	Mean	116.4624694	55.9547262	115.942776	52.6112894
3000	**Best**	38.80334777	4.001886266	51.7377346	0.065497359
	Worst	149.3852153	98.57159506	159.192274	134.3184949
	Mean	100.2171148	103.0026754	97.9436946	61.0832188

Table 4. *Cont.*

Rastrigin Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
4000	**Best**	54.72265233	52.73263809	39.7980992	47.75788823
	Worst	156.2075185	144.4814676	134.389914	135.4559257
	Mean	101.1418009	90.6758531	99.6304181	94.61300534
5000	**Best**	66.73304166	36.81341449	75.6164833	53.7274902
	Worst	156.2788002	146.4002774	144.339083	138.369496
	Mean	101.0967699	93.80047281	109.224593	91.73710009

Table 5 shows the results of executing the proposed ICA algorithm 30 times with dynamic adaptation of the β and ξ parameters, where we find the average, best and worst results obtained for the Griewank function.

Table 5. Griewank function.

Griewank Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	5.37×10^{-6}	0.00044923	0.00180682	0.00152579
	Worst	1.02242034	1.03634946	1.02883464	1.03790429
	Mean	0.35910253	0.50333873	0.41852252	0.31999397
2000	**Best**	5.70×10^{-6}	0.00112137	0.00063138	0.00013872
	Worst	1.04181924	1.04220393	1.05562693	1.02499176
	Mean	0.34491885	0.4949299	0.36873076	0.28444478
3000	**Best**	0.00021101	0.00250878	0.00568237	0.01177726
	Worst	0.8733779	1.04327883	1.03165716	1.02931357
	Mean	0.25614264	0.50039122	0.4444043	0.30456505
4000	**Best**	0.01693866	0.00080968	0.00021682	8.37×10^{-5}
	Worst	1.02590811	1.03539577	1.04268116	1.02265887
	Mean	0.46998072	0.46775839	0.33053626	0.28910338
5000	**Best**	0.00052846	0.00659022	9.13×10^{-5}	0.00086851
	Worst	1.03574438	1.03888314	1.03172748	1.02819675
	Mean	0.53988701	0.66356327	0.29744525	0.34420119

Table 6 shows the results of executing 30 times the proposed ICA algorithm with dynamic adaptation the β and ξ parameters, where we find the average, best and worst results obtained for the Ackley function.

Table 6. Ackley function.

Ackley Function					
Decades		**ICA**	**β Inc**	**ξ Inc**	**β Inc–ξ Dec**
			1–2	**0–1**	**1–2 and 0–1**
1000	**Best**	0.04282997	0.1818607	0.98406118	0.20184042
	Worst	9.59713997	7.80621452	9.90372665	10.6229277
	Mean	5.00997147	4.6910324	5.53751731	4.67171762
2000	**Best**	1.62876913	0.06634036	1.16270802	0.15545381
	Worst	8.25352738	6.99145708	10.2785221	9.83562472
	Mean	5.03586405	2.91434453	5.28942225	4.73526718
3000	**Best**	0.07369281	0.17195685	0.01459694	1.40667014
	Worst	9.65566678	4.63766418	10.5080894	9.66222211
	Mean	5.62259386	2.91120991	4.86681416	5.48898051
4000	**Best**	0.01183355	0.03355728	0.44424216	0.14476083
	Worst	8.68429605	5.65397345	11.1485964	10.5471553
	Mean	4.69447289	2.37875375	5.18774309	4.97000206
5000	**Best**	0.07598449	0.00936163	1.39061542	0.01190486
	Worst	9.98139202	6.69937633	8.39877466	9.53387625
	Mean	5.25589516	3.2539677	5.137672	4.17327678

5. Statistical Comparison for ICA and FICA with β Increasing

The statistical test was performed between the ICA algorithm and the FICA algorithm with dynamic adjustment of the β parameter with the range from 1 to 2, and the experiments correspond to the results obtained in tests for the ICA and FICA algorithms. Each algorithm was applied separately to each of the mathematical functions, 6 functions were used for the algorithms and 30 experiments were realized. Table 7 contains the average and standard deviation obtained after 30 executions for algorithm with 1000 decades. The values of parameters utilized in the fuzzy imperialist competitive algorithm and the ICA Algorithm [24,31] are the following:

- No. of Countries: 200.
- No. of Imperialists: 10.
- Revolution Rate: 0.2.
- β: 1.4.
- ξ: 0.02.
- Dimensions: 30.
- Simulations: 30.
- Decades: 1000.

Table 7 shows the performance of the FICA algorithm with dynamic adjustment of the β parameter and the ICA algorithm for 30 times we find the mean and standard deviation results.

The statistical test used for the comparison is the Wilcoxon test [32] that is commonly utilized to analyze data, where its parameters are shown in Table 8.

Table 7. Comparison with ICA and FICA with increasing β.

Function		ICA	FICA
Sphere	**Mean**	2.51×10^{-21}	2.27×10^{-25}
	S.D.	1.37×10^{-20}	4.17×10^{-25}
Quartic	**Mean**	9.75×10^{-41}	2.96×10^{-39}
	S.D.	5.34×10^{-40}	1.58×10^{-38}
Griewank	**Mean**	0.3591025	0.5033387
	S.D.	0.3903724	0.3845768
Rosenbrock	**Mean**	18.32843	17.302077
	S.D.	5.6853778	5.9223444
Rastrigin	**Mean**	131.0165	95.81005
	S.D.	22.167742	29.295518
Ackley	**Mean**	5.0099715	4.6910324
	S.D.	2.2992776	1.8764895

Table 8. Parameters for the statistical test.

Function	No.	F1	F2	Difference	Abs (Difference)	Rank	Signed Rank (−)	Signed Rank (+)
		ICA	FICA(β)					
Spherical	1	2.51×10^{-21}	2.27×10^{-25}	2.51×10^{-21}	2.51×10^{-21}	2		2
Quartic	2	9.75×10^{-41}	2.96×10^{-39}	-2.86×10^{-39}	2.86×10^{-39}	1	1	
Griewank	3	0.35910253	0.50333873	−0.14423619	0.14423619	3	3	
Rosenbrock	4	18.3284301	17.3020772	1.02635282	1.02635282	5		5
Rastrigin	5	131.016497	95.8100497	35.2064469	35.2064469	6		6
Ackley	6	5.00997147	4.6910324	0.31893908	0.31893908	4		4

The null hypothesis tells us that the average of the results obtained by the fuzzy imperialist competitive algorithm is equal to the average of the ICA algorithm, and the alternative hypothesis states that the average of the results obtained by the fuzzy imperialist competitive algorithm is different to the average performance of the ICA algorithm.

To test the hypothesis, the absolute values $|Z_i| \ldots |Z_n|$ are ordered and assigned its range on Rank, the order of these values are from lowest to highest, Sign (+) column indicates the values are positive and Sign (−) column indicates the values are negative.

The equation used in the statistical test is:

$$W^+ = \sum_{\approx i>0} R_i \tag{19}$$

The sum of the ranges R_i corresponding to the positive values of Z_i.

W^+ represent the value of the sum of the positive ranks, W^- is the value of the sum of the negative ranks, W represent the differences among two samples, and the value of the $W0$ represents the value for a two-tailed test in the table utilizing 30 samples.

The test of Wilcoxon to evaluate is the one shown below

If $W \leq W0$

Then fail to reject H_0

Table 9 shows a statistical test performed for two methods with a confidence level of 95% and a value of $W = 4$ and $W0 = 1$. The results obtained from the statistical test are as follows: for the fuzzy imperialist competitive algorithm, it fails to reject H_0 the null hypothesis, therefore the alternative hypothesis is rejected, which means that the average performance of the fuzzy imperialist competitive algorithm is not significantly different to the average performance of the ICA algorithm.

Table 9. Parameters for the statistical test.

W^-	W^+	W	Level Significance	m = Degrees of Freedom	$W0 = W\alpha, m$
4	17	4	0.05	6	1

6. Result of the Experiments and Statistical Test of the Fuzzy Imperialist Competitive Algorithm (FICA) and Fuzzy Cuckoo Search (FCS)

In this section the fuzzy imperialist competitive algorithm (FICA) and the Fuzzy Cuckoo search Algorithm (FCS) are compared with 5 mathematical functions with 16 dimensions for the β parameter and a combination of β-ξ, and the Pa parameter the in the Fuzzy Cuckoo search (FCS). The results obtained with the FICA algorithm and the FCS are shown in separate tables for each function. The tables contain the average values obtained after 50 executions for algorithm with decades 2000 and 2000 iterations respectively.

The parameters used in the fuzzy imperialist competitive algorithm and the FCS Algorithm [33] are the following:

- No. of Countries: 200 (FICA).
- No. of Imperialists: 10 (FICA).
- Revolution Rate: 0.2 (FICA).
- Dimensions: 16.
- Simulations: 50.
- Decades: 2000.
- Population: 25 nests (FCS).
- β: 1.5 (FCS).

Table 10 shows the results of the algorithm FICA and the FCS algorithm executed 50 times, dynamically adapting the β, ξ, and Pa parameters, and we show the average of the results obtained.

Table 10. Comparison with FICA and Fuzzy Cuckoo Search (FCS) (pa).

Function	β Inc	ξ Inc	β Inc–ξ Dec	FCS
	1–2	0–1	1–2 and 0–1	Pa
Spherical	5.16×10^{-130}	1.06×10^{-164}	1.93×10^{-128}	2.23×10^{-40}
Griewank	9.02×10^{-2}	2.90×10^{-1}	1.68×10^{-1}	6.06×10^{-12}
Rosenbrock	1.68×10^{-1}	1.67×10^{-1}	2.89×10^{-2}	2.39×10^{-5}
Rastrigin	7.2952	23.8532	7.3277	11.6309
Ackley	4.5930	4.2932	3.5919	7.47×10^{-14}

The statistical test that was used for the comparison is the Wilcoxon paired test and the data and parameters are shown in Tables 11–14.

Table 11. Parameters for the statistical test for FICA (β) and FCS (pa).

		F1	F2					
Function	**No.**	**FCS (pa) [33]**	**FICA (β)**	**Difference**	**Abs (Difference)**	**Rank**	**Signed Rank (−)**	**Signed Rank (+)**
Spherical	1	2.23×10^{-40}	5.16×10^{-130}	2.23×10^{-40}	2.23×10^{-40}	1		1
Griewank	3	6.06×10^{-12}	9.02×10^{-2}	−0.09025	0.09025	2	2	
Rosenbrock	4	2.39×10^{-5}	1.68×10^{-1}	−0.1686561	0.1686561	3	3	
Rastrigin	5	11.6309	7.2952	4.3357	4.3357	4		4
Ackley	6	7.47×10^{-14}	4.593	−4.593	4.593	5		5

The null hypothesis tells us that the average of the results obtained by the FICA (β) algorithm is equal to the average of the FCS (pa) algorithm and the alternative hypothesis states that the average of the obtained by the FICA (β) algorithm is different to the average performance of the FCS (pa) algorithm.

Table 12 shows a statistical test performed for two methods with a confidence level of 95% and a value of $W = 5$ and $W0 = 1$. The results obtained from the statistical test are as follows: for the FICA algorithm, it fails to reject H_0 the null hypothesis and the alternative hypothesis is rejected, which means that the average performance of the FICA algorithm is not significantly different to the performance of the FCS (pa) algorithm.

Table 12. Parameters for the statistical test.

W^-	W^+	W	Level Significance	m = Degrees of Freedom	$W0 = W\alpha, m$
5	10	5	0.05	5	1

Table 13. Parameters for the statistical test for FICA combination (β-ξ) and FCS (pa).

Function	No.	F1	F2	Difference	Abs (Difference)	Rank	Signed Rank (−)	Signed Rank (+)
		FCS (pa) [33]	FICA(β-ξ)					
Spherical	1	2.23×10^{-40}	1.93×10^{-128}	2.23×10^{-40}	2.23×10^{-40}	1		1
Griewank	3	6.06×10^{-12}	1.68×10^{-1}	−0.1687	0.1687	3	−3	
Rosenbrock	4	2.39×10^{-5}	2.89×10^{-2}	−0.0289451	0.0289451	2	−2	
Rastrigin	5	11.6309	7.3277	4.3032	4.3032	5		5
Ackley	6	7.47×10^{-14}	3.5919	−3.5919	3.5919	4		4

The null hypothesis tells us that the average of the results obtained by the FICA combination (β-ξ) algorithm is equal to the average of the FCS (pa) algorithm and the alternative hypothesis states that the average of the results obtained by the FICA combination (β-ξ) algorithm is different to the average performance of the FCS (pa) algorithm.

Table 14 shows a statistical test performed for two methods with a confidence level of 95% and a value of $W = 5$ and $W0 = 1$. The results obtained from the statistical test are as follows: for the FICA combination (β-ξ) algorithm, it fails to reject H_0 the null hypothesis and the alternative hypothesis is

rejected, which means that the average performance of the FICA combination (β-ξ) algorithm is not significantly different to the average performance of the FCS (pa) algorithm.

Table 14. Parameters for the statistical test.

W^-	W^+	W	Level Significance	m = Degrees of Freedom	$W0 = W\alpha, m$
5	10	5	0.05	5	1

7. Conclusions

The ICA algorithm is a newly developed method based on the theory of socio-political developments, which can be used for both linear and nonlinear functions. Recently, the parameters of metaheuristic algorithms have mostly been chosen by trial and error, which is why we decided to carry out a modification to the ICA algorithm to achieve better performance than the original algorithm and improve convergence, in this way avoiding getting stuck in local optima.

This research presents a modification of the imperialist competitive algorithm to enable dynamic adaptation of parameters utilizing fuzzy logic, this in order that the fuzzy system dynamically moves the ICA algorithm parameters. Firstly, the ICA algorithm was analyzed and each of the parameters used in processing, and then experiments with different ranges and possible values that each parameter could take were analyzed and then the algorithm was applied to Benchmark mathematical functions in order to optimize such functions and after that for comparing with other existing metaheuristic methods and in this way measure how good our method is.

We designed different fuzzy systems to independently move each of the parameters of the ICA algorithm, since by these parameters the exploitation and exploration of the algorithm are dynamically controlled. We designed and implemented different fuzzy system combinations between parameters and every one of the modifications was applied to the benchmark mathematical functions.

Finally, after analyzing the results of each of the proposals, we conclude that the fuzzy imperialist competitive algorithm is better than the original algorithm on average; however, we cannot affirm this statistically since our statistical test yield shows that the results are very close. In addition, it was not possible to significantly improve the results of the fuzzy Cuckoo search algorithm and we only managed to improve on some of the reference functions using the dynamic adjustment of the parameters individually, because with the proposed approach it was not possible to get significantly better results.

After studying the original imperialist competitive algorithm and making the modification using fuzzy logic to dynamically adjust the parameters, we can conclude that the ICA algorithm is good for solving complex optimization problems, in which we can even achieve better results by applying the following actions, which can be considered as future work:

- Use different numbers of dimensions.
- Optimize the rules on the combined method to see if better results are achieved.
- Use different inputs in the fuzzy system.
- Change the fuzzy systems to be of type-2 to better model uncertainty.
- Apply the different methods to control problems.
- Perform hybridization with other metaheuristic algorithms for improving the ICA algorithm.

Author Contributions: Fevrier Valdez and Jose Soria reviewed the state of the art; Oscar Castillo contributed to the discussion and analysis of the results; Emer Bernal analyzed of the original method and used fuzzy logic for parameter adaptation, contributed to the simulations and wrote the paper. All authors have read and approved the final manuscript.

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

References

1. Goldansaz, S.M.; Fariborz, J.; Zahedi Anaraki, A.H. A hybrid imperialist competitive algorithm for minimizing makespan in a multi-processor open shop. *Appl. Math. Model.* **2013**, *37*, 9603–9616. [CrossRef]
2. Olafsson, S. Metaheuristics. In *Handbooks in Operations Research and Management Science*; Elsevier: Amsterdam, The Netherlands, 2006; Volume 13, pp. 633–654.
3. Greenhalgh, D.; Marshall, S. Convergence Criteria for Genetic Algorithms. *SIAM J. Comput.* **2000**, *30*, 269–282. [CrossRef]
4. Woddis, J. *An Introduction to Neo-Colonialism*; Lawrence and Wishart: London, UK, 1967.
5. Zhou, W.; Yan, J.; Li, Y.; Xia, C.; Zheng, J. Imperialist competitive algorithm for assembly sequence planning. *Int. J. Adv. Manuf. Technol.* **2012**, *67*, 2207–2216. [CrossRef]
6. Duan, H.; Huang, L.Z. Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning. *Neurocomputing* **2013**, *125*, 166–171. [CrossRef]
7. Lucas, C.; Nasiri-Gheidari, Z.; Tootoonchian, F. Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Convers. Manag.* **2010**, *51*, 1407–1411. [CrossRef]
8. Sarayloo, F.; Tavakkoli-Moghaddam, R. Imperialistic Competitive Algorithm for Solving a Dynamic Cell Formation Problem with Production Planning. *Adv. Intell. Comput. Theor. Appl.* **2010**, *6215*, 266–276.
9. Dorigo, M.; Blum, C. Ant Colony Optimization theory: A Survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278. [CrossRef]
10. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1999.
11. Bontoux, B.; Feillet, D. Ant colony optimization for the traveling purchaser problem. *Comput. Oper. Res.* **2008**, *35*, 628–637. [CrossRef]
12. Engelbrecht, A.P. *Computational Intelligence*; Wiley: Pretoria, South Africa, 2007.
13. Gen, M.; Yun, Y. Soft computing approach for reliability optimization: State-of-the-art survey. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 1008–1026. [CrossRef]
14. Jang, J.-S.; Sun, C.-T.; Mizutani, E. *Neuro-Fuzzy and Soft Computing a Computational Approach to Learning and Machine Intelligence*; Prentice Hall: Upper Saddle River, NJ, USA, 1997.
15. Hosseini, S.M.; Al Khaled, A. A survey on the Imperialist Competitive Algorithm Metaheuristic: Implementation in Engineering Domain and Directions for Future Research. *Appl. Soft Comput. J.* **2014**, *24*, 1078–1094. [CrossRef]
16. Atashpaz-Gargari, E.; Hashemzadeh, F.; Rajabioun, R.; Lucas, C. Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process. *Int. J. Intell. Comput. Cybern.* **2008**, *1*, 337–355. [CrossRef]
17. Lian, K.; Zhang, C.; Gao, L.; Shao, X. A modified colonial competitive algorithm for the mixed-model U-line balancing and sequencing problem. *Int. J. Prod. Res.* **2012**, *50*, 5117–5131. [CrossRef]
18. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm for minimum bit error rate beamforming. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 125–133.
19. Rasul, E.; JavedaniSadaei, H.; Abdullah, A.H.; Gani, A. Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (RHWFTS–ICA) for short term load forecasting. *Energy Convers. Manag.* **2013**, *76*, 1104–1116.
20. Shamshirband, S.; Amini, A.; Nor Badrul, A.; Mat Kiah, M.L.; Ying, W.T.; Furnell, S. D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *J. Int. Meas. Confed.* **2014**, *55*, 212–226. [CrossRef]
21. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, 25–28 September 2007; pp. 4661–4667.
22. Nourmohammadia, A.; Zandiehb, M.; Tavakkoli-Moghaddamca, R. An imperialist competitive algorithm for multi-objective U-type assembly line design. *J. Comput. Sci.* **2012**, *4*, 393–400. [CrossRef]
23. Banaei, M.; Seyed-Shenava, S.; Farahbakhsh, P. Dynamic stability enhancement of power system based on a typical unified power flow controllers using imperialist competitive algorithm. *Ain Shams Eng. J.* **2014**, *5*, 691–702. [CrossRef]

24. Hadidi, A.; Hadidi, M.; Nazari, A. A new design approach for shell-and-tube heat exchangers using imperialist competitive algorithm (ICA) from economic point of view. *Energy Convers. Manag.* **2013**, *67*, 66–74. [CrossRef]
25. Valdez, F.; Melin, P.; Castillo, O. An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. *Soft Comput.* **2011**, *11*, 2625–2632. [CrossRef]
26. Haunpt, R.L.; Haunpt, S.E. *Practical Genetic Algorithms*, 2nd ed.; Wiley and Sons: Hoboken, NJ, USA, 2004.
27. 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 Syst. Appl.* **2012**, *40*, 3196–3206. [CrossRef]
28. Dallegrave Afonso, L.; Cocco Mariani, V.; dos Santos Coelho, L. Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. *Expert Syst. Appl.* **2013**, *40*, 3794–3802. [CrossRef]
29. Duan, H.; Xu, C.; Liu, S.; Chao, S. Template matching using chaotic imperialist competitive algorithm. *Pattern Recognit. Lett.* **2010**, *31*, 1868–1875. [CrossRef]
30. Mozafaria, H.; Behzad, A.; Ayoba, A. Optimization of Adhesive-Bonded Fiber Glass Strip using Imperialist Competitive Algorithm. *Procedia Technol.* **2012**, *1*, 194–198. [CrossRef]
31. Jula, A.; Othman, Z.; Sundararajan, E. Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition. *Expert Syst. Appl.* **2014**, *42*, 135–145. [CrossRef]
32. Peraza, C.; Valdez, F.; Garcia, M.; Melin, P.; Castillo, O. A New Fuzzy Harmony Search Algorithm Using Fuzzy Logic for Dynamic Parameter Adaptation. *Algorithms* **2016**, *9*, 69. [CrossRef]
33. Guerrero, M.; Castillo, O.; Garcia, M. Fuzzy dynamic parameter adaptation in the Cuckoo Search Algorithm via Lévy flights for optimizing benchmark mathematical functions. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 441–448.