JOURNAL OF UNIVERSAL MATHEMATICS

VOL.2 NO.2 pp.183-207 (2019)

ISSN -2618-5660

DIFFERENTIAL EVOLUTION WITH DYNAMIC ADAPTATION OF PARAMETERS BASED ON A FUZZY LOGIC AUGMENTATION APPROACH

Patricia Ochoa, Oscar Castillo*, José Soria

ABSTRACT. This paper proposes an improvement to the Differential Evolution algorithm using a fuzzy logic augmentation. The main contribution is to dynamically adapt the parameters of mutation (F) and crossover (CR) using a fuzzy system, with the aim that the fuzzy system calculates the optimal parameters of the differential evolution algorithm during execution for obtaining better solutions, in this way arriving to the proposed new fuzzy differential evolution algorithm. In this paper experiments are performed with a set of mathematical functions using the original algorithm and the proposed method. Based on a statistical comparison of the original and proposed method, we can state that the fuzzy differential evolution algorithm outperforms the original differential evolution method.

1.INRODUCTION

Recently the use of fuzzy logic in evolutionary computing is becoming a common approach to improve the performance of the algorithms [20] [25] [26]. In most of the cases in the literature the parameters involved in the algorithms are determined by trial and error. In this aspect we propose the application of fuzzy logic, which can then be responsible of performing the dynamic adjustment of the mutation and crossover parameters in the Differential Evolution (DE) algorithm. This has the goal of providing better performance to Differential Evolution with a fuzzy logic augmentation of this algorithm.

Fuzzy logic or multi-valued logic is based on the fuzzy set theory proposed by Zadeh in 1965, which can help us with modeling expert knowledge, through the use of if-then fuzzy rules. Fuzzy set theory provides a systematic calculus to deal with linguistic information, and improves the numerical computation by using linguistic labels stipulated by membership functions [12]. Differential Evolution (DE) is one of the latest evolutionary algorithms that have been proposed. It was created in 1994 by Price and Storn in an attempt to solve the Chebychev polynomial problem. The following years these two authors also proposed the DE for optimization of nonlinear and non-differentiable functions on continuous spaces.

The DE algorithm is a direct search stochastic method, which has proven to be effective, efficient and robust in a wide variety of applications, such as the learning of a neural network, optimal filter design, and aero dynamical optimization. The DE has a number of important features, which makes it attractive for solving global optimization problems, among them are the following: it has the ability to handle non-differentiable, nonlinear and multimodal objective functions, and usually converges to the optimal solution with few control parameters, etc.

Keywords and phrases. Differential Evolution, Fuzzy Logic, Dynamic Parameters.

The DE belongs to the class of evolutionary algorithms that is based on populations. It uses two evolutionary mechanisms for the generation of descendants: mutation and crossover; finally a replacement mechanism, which is applied between the father vector and son vector determining who survives into the next generation. There exist works, where they are currently using fuzzy logic to optimize the performance of metaheuristic algorithms, to name a few, papers such as: [1], [2], [4], [6], [7], [31], [8],[10], [29], [22], [27] and [30].

Similarly, there are papers on Differential Evolution (DE) applications that use this algorithm to solve real world problems. To mention a few: [5], [11], [3], [13], [18], [20],[21], [24],[26],[33], [19] and [21].

The main contribution of this paper is the proposed Fuzzy Differential Evolution approach that is based on using fuzzy systems to dynamically adapt the parameters of the DE algorithm to improve the exploration and exploitation abilities of the method. The proposed Fuzzy Differential Evolution approach is different from existing works in the literature and for this reason is the main contribution of this paper.

This paper is organized as follows: Section 2 shows the concept of the Differential Evolution algorithm. Section 3 describes the proposed methods. Section 4 outlines the Benchmark Functions, Section 5 shows experiments with the Differential Evolution algorithm varying F (mutation parameter), Section 6 shows experiments with the algorithm of Differential Evolution varying CR (crossover parameter), Section 7 presents a fuzzy system for dynamic change of F and Cr, Section 8 shows the Statistical Tests, Section 9 shows the Wilcoxon test statistics and Section 10 offers theConclusions.

2.DIFFERENTIAL EVOLUTION

Differential Evolution (DE) is an optimization method belonging to the category of evolutionary computation that can be applied in solving complex optimization problems.

The DE is composed of 4 basic steps: initialization, mutation, crossover, selection.

This is a non-deterministic technique based on the evolution of a vector population (individuals) of real values representing the solutions in the search space. The generation of new individuals is carried out by the differential crossover and mutation operators [16].

The operation of the algorithm is explained below:

Population structure

The Differential Evolution algorithm maintains a pair of vector populations, both of which contain Np D-dimensional vectors of real-valued parameters [17].

$$P_{x,g} = (\mathbf{x}_{i,g}),\ i=0,1, \ldots , Np,\ \ g=0,1, \ldots, g_{max} \qquad (1)$$

$$\mathbf{x}_{i,g} = (x_{j,i,g}),\ j=0,1, \ldots, D-1 \qquad (2)$$

$$P_{v,g} = (\boldsymbol{v}_{i,g}),\ i=0,1, \ldots, Np-1,\ g=0,1, \ldots, g_{max} \qquad (3)$$

$$\boldsymbol{v}_{i,g} = (v_{j,I,g}),\ \ j=0,1, \ldots, D-1 \qquad (4)$$

Each vector in the current population is recombined with a mutant vector to produce a trial population, P_u, mutant vector $\mathbf{u}_{i,g}$:

$$P_{v,g} = (\boldsymbol{u}_{i,g}),\ i=0,1,\ \ldots,\ Np\text{-}1,\ g=0,1,\ \ldots,\ g_{max} \tag{5}$$

$$\boldsymbol{u}_{i,g} = (u_{j,I,g}),\ \ j=0,1,\ \ldots,\ D\text{-}1 \tag{6}$$

Initialization

Before initializing the population, the upper and lower limits for each parameter must be specified. These 2D values can be collected by two initialized vectors, D-dimensional, b_L and b_U, for which the subscripts L and U indicate the lower and upper limits respectively. Once the initialization limits have been specified a number generator randomly assigns each parameter in every vector a value within the set range. For example, the initial value (g = 0) of the j-th vector parameter is i-th:

$$x_{j,i,0} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \tag{7}$$

Mutation

In particular, the differential mutation uses a random sample equation showing how to combine three different vectors chosen randomly to create a mutant vector.

$$\boldsymbol{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \tag{8}$$

The scale factor, F ∈ (0,1) is a positive real number that controls the rate at which the population evolves. While there is no upper limit on F, the values are rarely greater than 1.0.

Crossover

To complement the differential mutation search strategy, DE also uses uniform crossover. This is sometimes known as discrete recombination (dual). In particular, DE crosses each vector with a mutant vector:

$$U_{i,g} = (u_{j,i,g}) = \left\{ \begin{array}{ll} v_{j,i,g} & if\left(rand_j(0,1) \le Cr\ or\ j=j_{rand}\right) \\ x_{j,i,g} & otherwise. \end{array} \right\} \tag{9}$$

Selection

If the test vector, $\boldsymbol{U}_{i,g}$ has a value of the objective function equal to or less than, its target vector, $\mathbf{X}_{i,g}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least another generation [16].

$$X_{i,g+1} = \begin{cases} U_{i,g} & if \ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g} & otherwise. \end{cases} \tag{10}$$

3.PROPOSED METHOD

The Differential Evolution (DE) Algorithm is a powerful search technique used for solving optimization problems. In this paper a new algorithm called Fuzzy Differential Evolution (FDE) with dynamic adjustment of parameters is proposed. The main goal is that the fuzzy system can dynamically provide to the algorithm with the optimal parameters during execution for the best performance of the DE algorithm.

We propose exploring the algorithm first by modifying in a separate fashion the mutation and crossover parameters respectively. This means having two fuzzy systems for each of the two parameters, which means that to dynamically modify the F parameter (mutation) we will have a fuzzy system that modifies F in an increase fashion and another fuzzy system that modifies F in a decrease fashion. In the same form we will design two fuzzy systems for the Cr (crossover) parameter that also changes in increment and decrement. Finally, we will build a fuzzy system which gives as output variables both the crossover and mutation parameters, with the idea that this fuzzy system can produce better results and to compare with other algorithms that have a fuzzy system as well as the algorithm that we proposed.
In this case the parameters that the fuzzy system optimizes are the crossover and mutation, as illustrated in Figure 1.

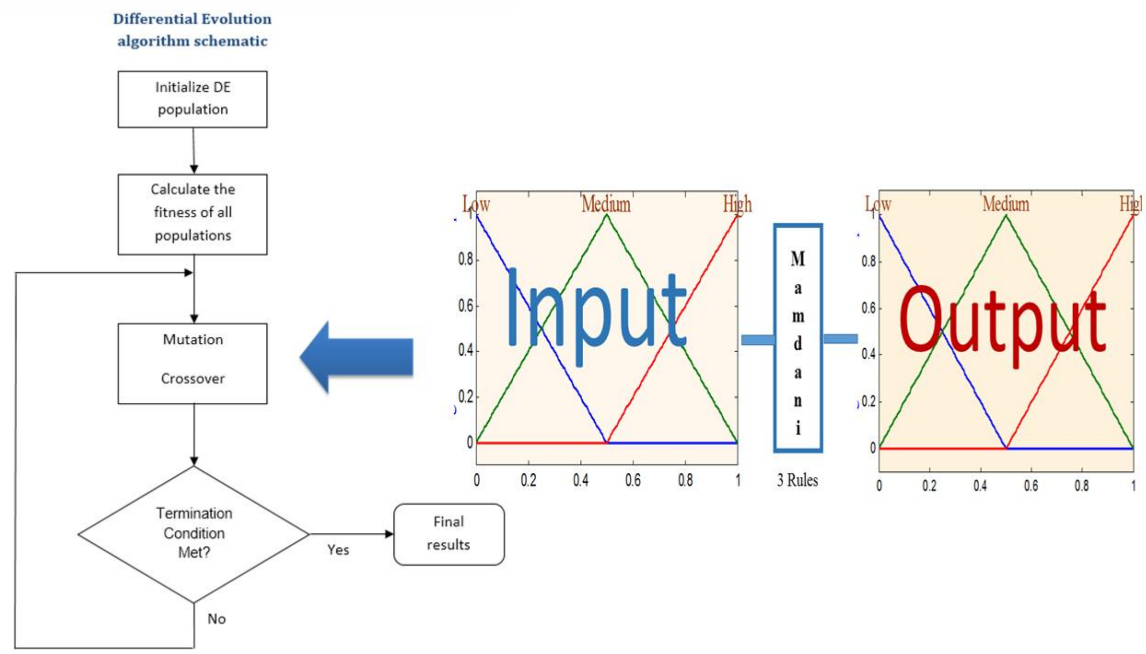

Fig. 1. The proposed Differential Evolution (DE) algorithm with a fuzzy system to dynamically adapt parameters.

4. BENCHMARK FUNCTIONS

In this paper, we consider 6 Benchmark functions for the tests, which are listed below [28] and illustrated in Figure 2.

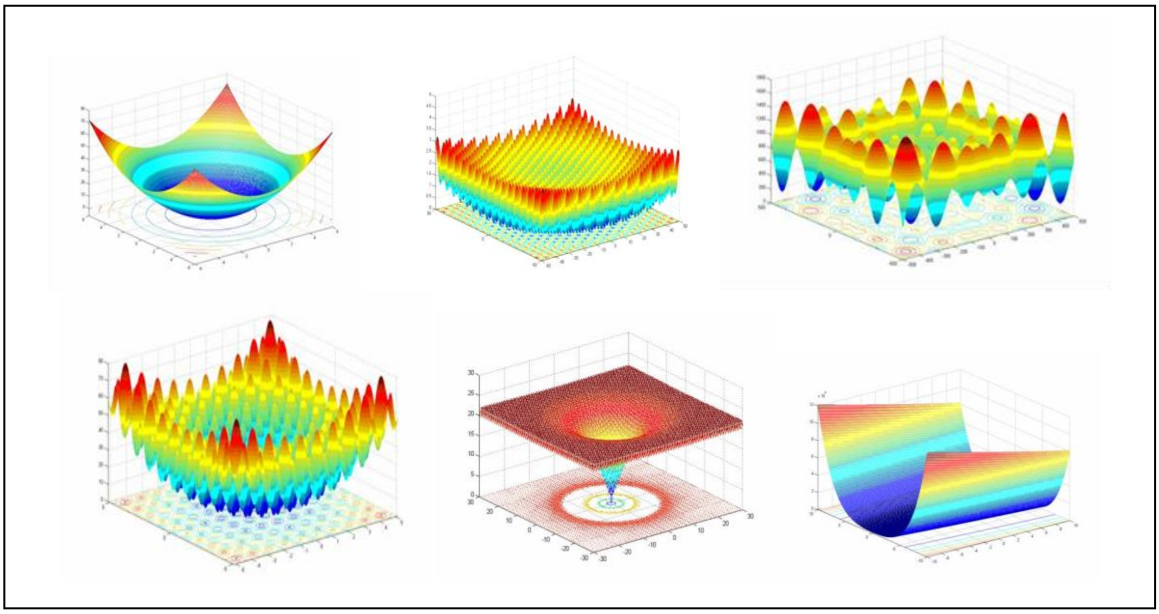

Fig.2 Benchmark mathematical functions

5.EXPERIMENTS WITH THE DIFFERENTIAL EVOLUTION ALGORITHM VARYING F (PARAMETER VARIABLE)

In this section we show simulation results for the Benchmark functions by changing manually the F parameter and then by dynamically changing F using a fuzzy system.

5.1 EXPERIMENTS VARYING F MANUALLY

We perform experiments with the of Differential Evolution algorithm for each function, by manually changing the F parameter in a range of 0.1 to 0.9 and performing 30 experiments for each value of F, in other words 30 experiments for F = 0.1, 30 experiments for 0.2 and up to F = 0.9, averages are obtained for each value of F. The generations vary in a range of 100 to 5000, and an overall average is obtained at the end. Table 1 shows the parameters used to perform the experiments where: NP is the number of population, D is the number of the dimension of the vector, CR is the crossing, F is the mutation, GEN is the number of generations, L is the lower limit and H is the upper limit.

Table 1 Parameters of functions.

Parameters
NP = 250
D = 50
CR = 0.1
GEN = 100 hasta 5000
L = -500
H = 500

For the Ackley and Rosenbrock functions we modified the search spaces, taking the values recommended in the literature, what we were looking for the functions used that would have equal parameters for the behavior of the Differential Evolution algorithm, but with these two functions we did not obtained good results and Table 2 shows the parameters used for the Ackley and Rosenbrock functions.

Table 2 Ackley and Rosenbrock function parameters.

Function Ackley	**Function Rosenbrock**
NP = 250	NP = 250
D = 50	D = 50
CR = 0.1	CR = 0.1
GEN = 100 hasta 5000	GEN = 100 hasta 5000
L = -32.768	L = - 2.048
H = 32.768	H = 2.048

Table 3 shows the averages obtained by generation for each function where the variable F is modified manually.

Table 3.Overall average by function modifying F manually.

Overall average by manually modifying F							
	Generations						
	100	500	1000	2000	3000	4000	5000
Sphere	2.75E+05	3.42E+03	3.91E+01	9.61E-03	2.39E-06	6.10E-10	1.54E-13
Griewank	6.99E+01	1.37E+00	2.07E-01	8.79E-04	2.22E-07	5.78E-11	1.52E-14
Schwefel	1.04E+04	8.10E+03	6.39E+03	4.04E+03	5.49E+02	1.29E-01	1.32E-01
Rastringin	2.78E+05	3.77E+03	2.25E+02	6.31E+01	4.37E+01	3.12E+01	1.66E+01
Ackley	1.36E+01	1.89E+00	2.77E-01	1.43E-03	2.07E-05	3.27E-07	5.11E-09
Rosenbrock	1.93E+02	3.91E+01	1.46E+01	5.46E-02	9.64E-06	1.77E-09	3.47E-13

5.2 FUZZY SYSTEM TO DYNAMICALLY MODIFY F

In previous experiments we realize that as more generations are used the results in the functions are better, the next step is to develop a fuzzy system that can help change the F parameter, and we have decided to develop a fuzzy system where the F parameter increases and another where F decreases. It is important to note that for the experiments, the parameters of Tables 1 and 6 are used for this set of functions.
This work considers two fuzzy systems with which the experiments were performed. It considers a fuzzy system which increases the F parameters and another that decreases the F parameter.

We first describe the fuzzy system, in which F is increased dynamically.

- Contains one input and one output
- Is of Mamdani type.

- All membership functions are triangular.
- The input of the fuzzy system is defined by the number of generations and is granulated into three membership functions and they are: MF1 = 'Low'[-0.5 0 0.5], MF2 = 'Medium' [0 0.5 1], MF3 = 'High'[0.5 1 1.5].
- The output of the fuzzy system is the F parameter is granulated in three membership functions which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The fuzzy system uses 3 rules and what it does is to increase the value of the F variable in a range of (0, 1).

The fuzzy rules are presented in Fig. 3.

1.- If (Generations is Low) then (F is Low)
2.- If (Generations is Medium) then (F is Medium)
3.- If (Generations is High) then (F is High)

Fig. 3 Rules of the fuzzy system.

Then the fuzzy system, in which F is dynamically decreased, is described as follows:

- Contains one input and one output
- Is of Mamdani type.
- All functions are triangular.
- The input of the fuzzy system is the number of generations and divided into three membership functions and they are: MF1 = 'Low'[-0.5 0 0.5], MF2 = 'Medium' [0 0.5 1], MF3 = 'High'[0.5 1 1.5].
- The output of the fuzzy system and the F parameter is divided in three membership functions which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The fuzzy system uses 3 rules and what it does is decreased the value of the F variable in a range of (0.1).

The fuzzy rules are presented in Fig. 4.

1.- If (Generations is Low) then (F is High)
2.- If (Generations is Medium) then (F is Medium)
3.- If (Generations is High) then (F is Low)

Fig. 4 Rules of the fuzzy system.

Experiments using the two proposed fuzzy systems where F dynamically increases and then decreases are performed. There are 30 experiments for each number of generations obtaining averages for each 30 experiments, where the generations range from 100 up to 5000, the value of F is dynamically changing between (0,1). Then we perform a comparison of the Benchmark functions with the Differential Evolution algorithm changing F manually, first with the fuzzy Differential Evolution (F in increase) and then with the fuzzy Differential Evolution (F in decrease).

Table 4 shows the comparison of results for the Sphere function, where the first column uses the general averages for each generation of Table 3 of the Differential Evolution algorithm.

Table 4. Simulation results of the Sphere function.

Sphere Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	2.75E+05	2.65E+05	1.92E+05
500	3.42E+03	3.59E+01	1.99E+01
1000	3.91E+01	4.82E-04	2.41E-04
2000	9.61E-03	7.29E-14	3.67E-14
3000	2.39E-06	1.32E-23	5.56E-24
4000	6.10E-10	1.87E-33	9.45E-34
5000	1.54E-13	3.20E-43	1.36E-43

A comparison of results of the Griewank function, where the first column uses the general averages for each generation of Table 3 of the Differential Evolution algorithm is presented in Table 5.

Table 5. Simulation results of the Griewank function.

Griewank Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	6.99E+01	6.71E+01	4.81E+01
500	1.37E+00	6.94E-01	5.14E-01
1000	2.07E-01	3.42E-05	1.84E-05
2000	8.79E-04	5.22E-15	2.28E-15
3000	2.23E-07	0.00E+00	0.00E+00
4000	5.79E-11	0.00E+00	0.00E+00
5000	1.53E-14	0.00E+00	0.00E+00

A comparison of results of the Schwefel function, where the first column uses the General averages for each generation of Table 3 of Differential Evolution algorithm is presented in Table 6.

Table 6. Simulation results of the Schwefel function.

Schwefel Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	1.04E+04	1.12E+04	1.09E+04
500	8.10E+03	5.19E+03	4.84E+03
1000	6.39E+03	1.45E+01	3.96E+00
2000	4.04E+03	6.36E-04	6.36E-04
3000	5.49E+02	6.36E-04	6.36E-04
4000	1.29E-01	6.36E-04	6.36E-04
5000	1.32E-01	6.36E-04	6.36E-04

A comparison of results of the Rastringin function, where the first column uses the general averages for each generation of Table 3 of the Differential Evolution algorithm is illustrated in Table 7.

Table 7. Simulation results of the Rastringin function.

Rastringin Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	2.78E+05	2.68E+05	1.85E+05
500	3.77E+03	3.79E+02	3.42E+02
1000	2.25E+02	1.44E+02	1.51E+02
2000	6.31E+01	7.64E+01	8.47E+01
3000	4.37E+01	4.75E+01	5.79E+01
4000	3.12E+01	2.42E+01	3.76E+01
5000	1.66E+01	1.22E-05	1.13E-04

A comparison of results of the Ackley function, where the first column uses the general averages for each generation of Table 3 of the Differential Evolution algorithm is summarized in Table 8.

Table 8. Simulation results of the Ackley function.

Ackley Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	1.36E+01	1.46E+01	1.33E+01
500	1.89E+00	4.36E-01	2.60E-01
1000	2.77E-01	8.44E-04	5.98E-04
2000	1.43E-03	1.05E-08	7.54E-09
3000	2.07E-05	1.30E-13	1.02E-13
4000	3.27E-07	8.94E-15	7.52E-15
5000	5.11E-09	7.99E-15	6.81E-15

A comparison of results of the Rosenbrock function, where the first column uses the general averages for each generation of Table 3 of the Differential Evolution algorithm is presented in Table 9.

Table 9. Simulation results of the Rosenbrock function.

Rosenbrock Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing F	Fuzzy Differential Evolution with Decreasing F
100	1.93E+02	1.66E+02	1.42E+02
500	3.91E+01	4.45E+01	3.44E+01
1000	1.46E+01	3.02E-01	1.55E+00
2000	5.46E-02	4.32E-12	2.21E-10
3000	9.64E-06	7.95E-23	2.02E-20
4000	1.78E-09	0.00E+00	0.00E+00
5000	3.47E-13	0.00E+00	0.00E+00

Comparing the values of Table 9 it can be noted that better results are obtained when using Fuzzy Differential Evolution with a Decreasing F parameter.

In Fig. 5 we show the convergence graphs of the 6 benchmark functions used in the study, where the original Differential Evolution algorithm, the Fuzzy Differential Evolution with increasing F and Fuzzy Differential Evolution with decreasing F are plotted, and we can clearly notice how both fuzzy DE variants outperform the traditional DE.

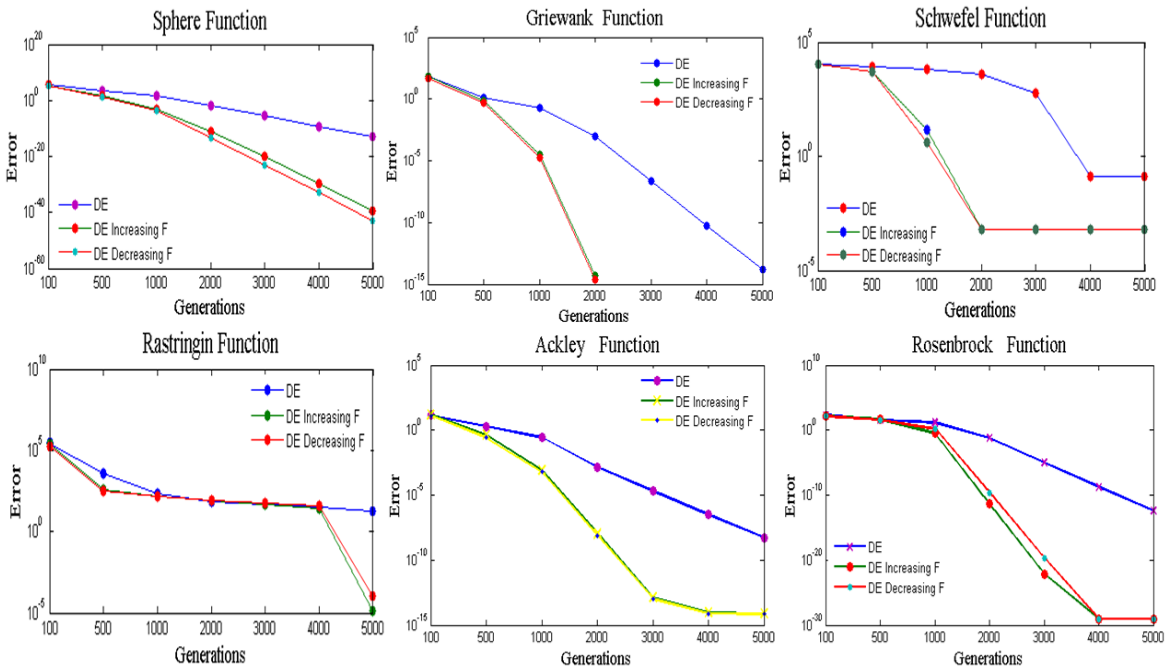

Fig. 5 Comparison for the set of Benchmark functions for the F parameter

6. EXPERIMENTATIONS WITH THE DIFFERENTIAL EVOLUTION ALGORITHM WITH A VARYING CR (CROSSOVER PARAMETER)

In this section we present results of the Fuzzy Differential method with dynamic changes in the crossover parameter.

6.1 EXPERIMENTS VARYING CR MANUALLY

For the experiments of the CR (crossover) parameter we use the same methodology with which we performed the experiments of the F parameter (mutation). We first performed the experiments with the traditional Differential Evolution algorithm by changing CR manually, and we use the same set of Benchmark functions for experiments with CR (crossover) and the parameters of Tables 1 and 2.

Table 11 shows the averages obtained by generation for each function where the variable CR parameter is modified manually.

Table 11.Overall average by function modifying CR manually.

Overall average by manually modifying CR							
	Generations						
Function	100	500	1000	2000	3000	4000	5000
Sphere	1.18E+04	1.60E+03	1.69E+03	1.98E+03	2.05E+03	2.00E+03	2.30E+03
Griewank	3.83E+00	5.65E-01	6.70E-01	5.30E-10	0.00E+00	0.00E+00	0.00E+00
Schwefel	8.37E+03	2.60E+02	1.10E+02	1.09E+02	1.17E+02	1.17E+02	2.28E+01
Rastringin	1.24E+03	4.32E+02	3.53E+02	3.78E+02	4.00E+02	3.19E+02	3.97E+02
Ackley	4.23E+00	6.75E-01	6.50E-01	6.07E-01	6.27E-01	6.10E-01	6.20E-01
Rosenbrock	3.40E+01	4.24E+00	3.76E+00	3.82E+00	3.73E+00	3.80E+00	3.90E+00

6.2 FUZZY SYSTEM TO DYNAMICALLY MODIFY CR

The experiments where we vary CR manually using the Differential Evolution algorithm do not show an improvement in our set of functions. The following experiments are now performed using the Fuzzy Differential Evolution algorithm, but now we dynamically change CR, as we did with F, and we perform this in two different ways of varying CR, in increment and decrement. It is important to note that for the experiments the parameters of Tables 15 and 20 are used for the set of functions.

This work considers two fuzzy systems with which the experiments are performed. A fuzzy system increases that the CR parameter and another that decreases the CR parameter dynamically in the algorithm.

We first describe the fuzzy system, in which CR is dynamically increased.

- Contains one input and one output
- Is of Mamdani type.
- All membership functions are triangular.
- The input of the fuzzy system is defined by the number of generations and granulated into three membership functions and they are: MF1 = 'Low'[-0.5 0 0.5], MF2 = 'Medium' [0 0.5 1], MF3 = 'High'[0.5 1 1.5].
- The output of the fuzzy system is the CR parameter and is granulated into three membership functions which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The fuzzy system uses 3 rules and what it does is to increase the value of the CR parameter in a range from 0 to 1.

The fuzzy rules are shown in Fig. 6.

1.- If (Generations is Low) then (CR is Low)
2.- If (Generations is Medium) then (CR is Medium)
3.- If (Generations is High) then (CR is High)

Fig. 6 Rules of the fuzzy system.

Then the fuzzy system, in which CR is dynamically decreased, is described as follows:

- Contains one input and one output
- Is Mamdani type.
- All functions are triangular.
- The input of the fuzzy system is the number of generations and it is divided into three membership functions and they are: MF1 = 'Low'[-0.5 0 0.5], MF2 = 'Medium' [0 0.5 1], MF3 = 'High'[0.5 1 1.5].
- The output of the fuzzy system and the F parameter is divided in three membership functions, which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The fuzzy system uses 3 rules and what it does is to decrease the value of the CR parameter in a range of 0 to 1.

The fuzzy rules are shown in Fig. 7.

1.- If (Generations is Low) then (CR is High)
2.- If (Generations is Medium) then (CR is Medium)
3.- If (Generations is High) then (CR is Low)

Fig. 7 Rules of the fuzzy system

Experiments using the two proposed fuzzy systems, where CR dynamically increases and then decreases were performed. There are 30 experiments for each case with different number of generations, obtaining averages for each case of 30 experiments, where the generations range from 100 up to 5000 and the value of CR is dynamically changing between 0 and 1.

Then we perform a comparison of the Benchmark functions with the Differential Evolution algorithm changing CR manually, the Fuzzy Differential Evolution (CR in increase) and the fuzzy Differential Evolution (CR in decrease).

Table 12 shows the comparison of results for the Sphere function, where the first column shows the general averages for each generation of Table 11 of the original Differential Evolution algorithm.

Table 12. Simulation results of the Sphere function.

Sphere Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	1.18E+04	1.38E+03	1.26E+03
500	1.60E+03	1.04E-05	6.48E+01
1000	1.69E+03	6.47E-07	2.43E+02
2000	1.98E+03	3.58E-17	6.52E+02
3000	2.05E+03	2.60E-11	3.84E+02
4000	2.00E+03	1.15E-18	5.51E+02
5000	2.36E+03	5.62E-05	7.36E+02

Table 13 shows the comparison of results of the Griewank function, where the first column shows the general averages for each generation of Table 11 of the Differential Evolution algorithm.

Table 13. Simulation results of the Griewank function.

Griewank Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	3.83E+00	1.37E+00	1.32E+00
500	5.65E-01	1.14E-06	4.65E-02
1000	6.70E-01	2.39E-08	2.57E-01
2000	5.30E-10	0.00E+00	4.30E-01
3000	0.00E+00	0.00E+00	5.77E-01
4000	0.00E+00	0.00E+00	6.97E-01
5000	0.00E+00	0.00E+00	8.98E-01

Table 14 summarizes the comparison of results of the Schwefel function, where the first column shows the general averages for each generation of Table 11 of the traditional Differential Evolution algorithm.

Schwefel Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	8.37E+03	1.11E+04	6.57E+03
500	2.60E+02	5.14E+03	3.91E-02
1000	1.10E+02	1.52E+01	6.36E-04
2000	1.09E+02	6.36E-04	6.36E-04
3000	1.17E+02	6.36E-04	6.36E-04
4000	1.17E+02	6.36E-04	6.36E-04
5000	1.23E+02	6.36E-04	6.36E-04

Table14. Simulation results of the Schwefel function.

Table 15 illustrates the comparison of results of the Rastringin function, where the first column shows the general averages for each generation of the Table 11 of the Differential Evolution algorithm.

Table 15. Simulation results of the Rastringin function

Rastringin Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	2.78E+05	2.68E+05	1.85E+05
500	3.77E+03	3.79E+02	3.42E+02
1000	2.25E+02	1.44E+02	1.51E+02
2000	6.31E+01	7.64E+01	8.47E+01
3000	4.37E+01	4.75E+01	5.79E+01
4000	3.12E+01	2.42E+01	3.76E+01
5000	1.66E+01	1.22E-05	1.13E-04

Table 16 presents the comparison of results of the Ackley function, where the first column shows the general averages for each generation of Table 11 of the Differential Evolution algorithm.

Table 16. Simulation results of the Ackley function

Ackley Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	4.23E+00	2.98E+00	2.79E+00
500	6.75E-01	3.37E-07	1.36E-01
1000	6.50E-01	6.09E-06	3.34E-01
2000	6.07E-01	8.77E-12	6.38E-01
3000	6.27E-01	4.22E-11	1.07E+00
4000	6.10E-01	3.43E-08	1.12E+00
5000	6.20E-01	4.44E-15	1.08E+00

Table 17 summarizes the comparison of results of the Rosenbrock function, where the first column uses the general averages for each generation of Table 11 of the Differential Evolution algorithm.

Table 17. Simulation results of the Rosenbrock function

Rosenbrock Function			
Generations	Differential Evolution	Fuzzy Differential Evolution with Increasing CR	Fuzzy Differential Evolution with Decreasing CR
100	3.40E+01	3.45E+01	2.22E+01
500	4.24E+00	3.30E-02	4.69E+00
1000	3.76E+00	8.12E-08	3.18E+05
2000	3.82E+00	3.30E-02	3.62E+05
3000	3.73E+00	0.00E+00	8.31E+00
4000	3.86E+00	3.90E-29	9.09E+00
5000	3.95E+00	0.00E+00	8.60E+00

In Fig. 8 we show the convergence graphs of the benchmark functions used in the experiments, where the original Differential Evolution algorithm, the Fuzzy Differential Evolution with increasing CR and the Fuzzy Differential Evolution decreasing CR are plotted. From this Figure it can be concluded that either one of the two fuzzy differential evolution variants outperforms the traditional DE algorithm.

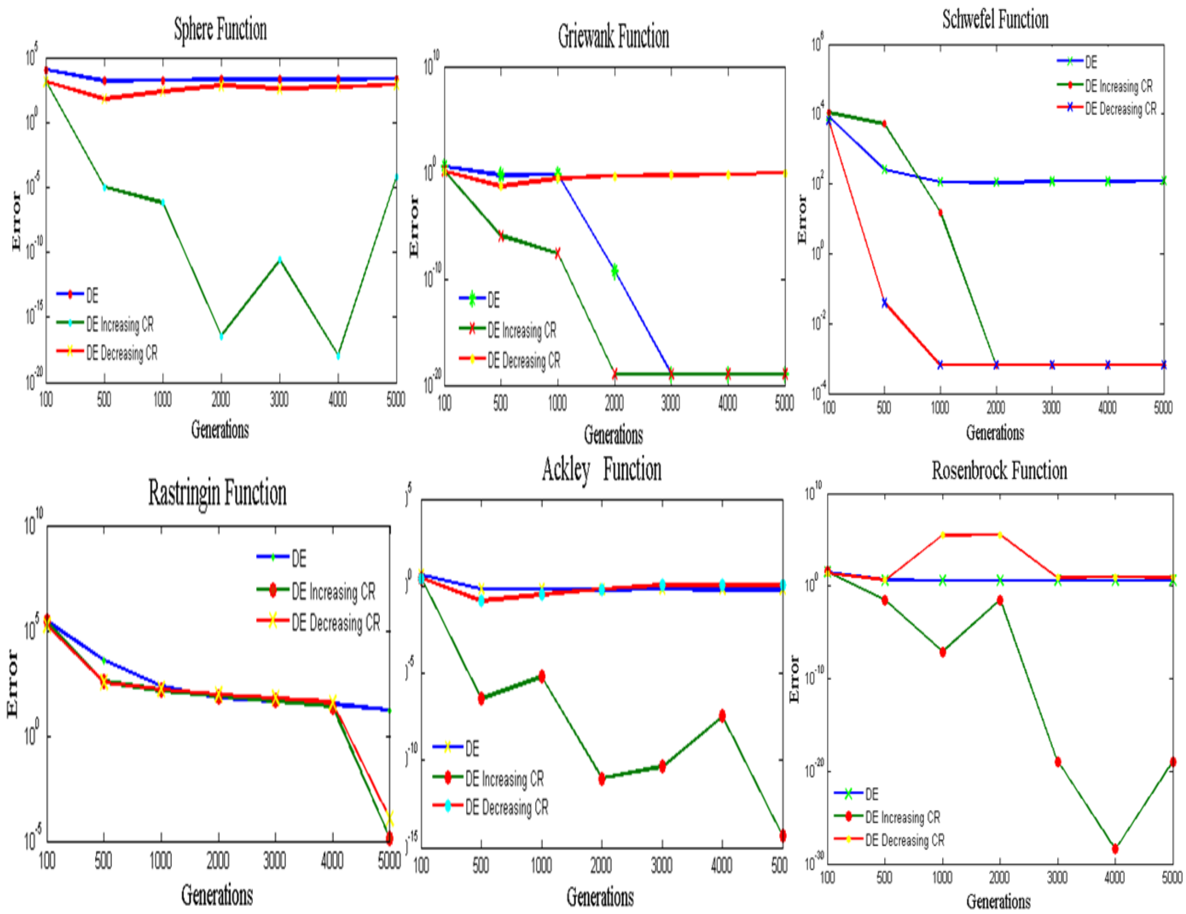

Fig.8 Comparison of the set of Benchmark functions for the CR parameter.

7.FUZZY SYSTEM FOR DYNAMIC ADAPTATION OF F AND CR

In this section we consider dynamically changing both the parameters in Fuzzy Differential Evolution, namely the F (mutation) and CR (crossing) parameters, at the time of executing the Differential Evolution algorithm.

Based on the experiments performed previously, where the F and CR parameters are changed separately by a fuzzy system, we decided to consider the form of the best results that we obtained for our set of benchmark functions. We consider the F parameter to change in decrease and for the CR parameter decided to build two fuzzy systems where, CR changes in increment and decrement, because in this parameter for certain functions is better to decrease and for others to increase, and with this we intend to obtain better results by having the two dynamic parameters working in simultaneously the fuzzy Differential Evolution algorithm.

The structure of the fuzzy system, where the F and CR parameters vary in a decrease fashion is as follows:

- Contains one input and two outputs and is of Mamdani type.
- All membership functions are triangular.
- The input of the fuzzy system is defined by the generations and is granulated into three membership functions and they are: MF1 = 'Low'[-0.5 0 0.5], MF2 = 'Medium' [0 0.5 1], MF3 = 'High'[0.5 1 1.5].
- The output of the fuzzy system corresponding to the F parameter is granulated into three membership functions, which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The output of the fuzzy system corresponding to the CR parameter is granulated into three membership functions, which are: MF1 = 'Low', [-0.5 0 0.5], MF2 = 'Medium', [0 0.5 1] MF3 = 'High', [0.5 1 1.5].
- The fuzzy system uses 3 rules and what it does is to decrease the value of the variables F and CR in the range (0, 1).

The fuzzy rules are shown in Fig. 9.

1.- If (Generations is Low) then (F is High)(CR is High)
2.- If (Generations is Medium) then(F is Medium) (CR is Medium)
3.- If (Generations is High) then (F is Low)(CR is Low)

Fig. 9 Rules of the fuzzy system

For the case of the fuzzy Differential Evolution algorithm that decreases the F and Cr parameters dynamically within a range of (0, 1), there are 30 experiments for each number of generations, where the generations range from 100 up to 5000. In this case for 100 generations 30 experiments are performed and the averages are obtained for each number of generations.
We performed experiments using the Fuzzy Differential Evolution algorithm, where the number of generations is used as the input and F and CR are used as outputs and these change in decrease, using the same set of Benchmark functions for experiments. The parameters to use for the first 4 functions Sphere, Griewank, Schwefel, Rastringin, are shown in Table 1 and for the Ackley and Rosenbrock functions the search space as shown in Table 2. Table 18 shows the results of the set of Benchmark functions we are using for the experiments and Fig. 10 represent the comparison.

Table 18. Results using the fuzzy system with two outputs F and CR in decrease

Fuzzy Differential Evolution algorithm(F and CR in decrease)						
Generations	Functions					
	Sphere	Griewank	Schwefel	Rastringin	Ackley	Rosenbrock
100	1.88E+05	4.64E+01	1.32E+04	1.93E+05	1.31E+01	1.61E+02
500	1.54E+02	1.03E+00	1.09E+04	5.83E+02	9.94E-01	6.35E+01
1000	2.50E-02	1.04E-03	9.64E+03	2.57E+02	6.88E-03	4.44E+01
2000	6.95E-10	2.64E-11	7.67E+03	2.03E+02	1.27E-06	2.29E+01
3000	2.08E-17	0.00E+00	6.09E+03	1.76E+02	2.12E-10	7.47E+00
4000	6.51E-25	0.00E+00	4.30E+03	1.61E+02	4.25E-14	5.02E-01
5000	1.84E-32	0.00E+00	2.24E+03	1.44E+02	6.34E-15	2.33E-02

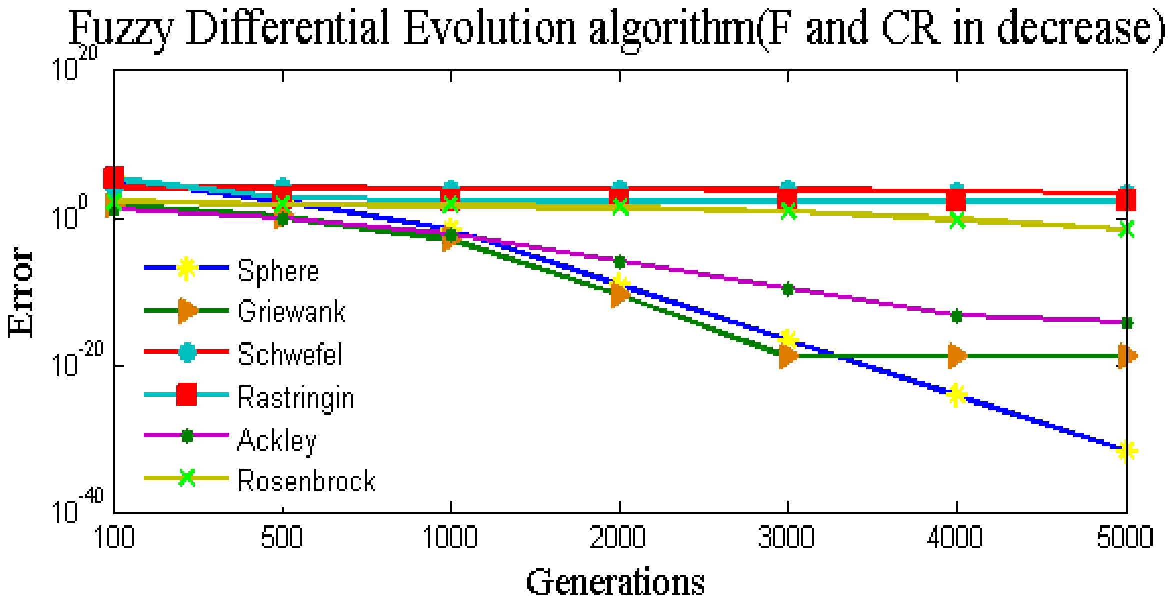

Fig. 10 Comparison of benchmark functions.

As we can note the results using the two parameters F and CR dynamically changing do not help to improve the results in the Fuzzy Differential Evolution algorithm, and we consider that the reason is that both parameter are not interacting in a correct way for improving performance in the differential evolution algorithm.

8.STATISTICAL TESTS

In this section we perform the statistical comparison of the original Differential Evolution algorithm with our proposed Fuzzy Differential Evolution method, and we perform the tests for the set of Benchmark functions used in this paper. The Fuzzy Differential Evolution algorithm that was considered is where the F (mutation) parameter changes dynamically in a decrease fashion, which gave us the better results in all functions used.

We perform statistical Z tests of two samples, and made a comparison between the original Differential Evolution algorithm and the proposed Fuzzy Differential Evolution, and this comparison is executed in the following manner:

Experiments with a manual Differential Evolution algorithm were performed with 5000 generations where F is changed manually, since the F parameter varies from 0.1 to 0.9 and for each F value there are 30 experiments, this gives us a total of 270 experiments of which we will take a random sample of 30. Fuzzy Differential Evolution algorithm, where 30 experiments were performed with the F parameter changing dynamically, therefore this is our sample to make the comparison.

The statistical test used for comparison is the z-test, whose parameters are defined in Table 20.

Table 20. Parameters for statistical testing

Parameter	Value
Level of significance	95%
Alpha	5 %
H_0	$\mu_1 \geq \mu_2$
H_a	$\mu_1 < \mu_2$(claim)
Critical value	-1.96

The null hypothesis states that the average of the Fuzzy Differential Evolution algorithm is greater than or equal to the average of the Differential Evolution algorithm, and on the other hand the alternative hypothesis states that the Fuzzy Differential Evolution algorithm average is lower than the average of the Differential Evolution algorithm, with a region of rejection for all values below - 1.96.
The equation for the test that was applied is as follows:

$$Z = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)}{\sigma_{\overline{X}_1 - \overline{X}_2}} \tag{11}$$

The data from the values of the means and standard deviations for the original method and the proposed method are in Table 21 for the set of benchmark functions used.

Table 21. Average and standard deviation values for the set of Benchmark functions

Original method			Proposed method		
Sphere	Mean	4.24E-12	Sphere	Mean	4.52E-53
	S.E.	6.75E-13		S.E	1.28E-53
Griewank	Mean	1.37467-13	Griewank	Mean	0.00E+00
	S.E.	3.28E-14		S.E	0.00E+00
Schwefel	Mean	3.26E-01	Schwefel	Mean	6.36E-04
	S.E.	5.24E-01		S.E	0.00E+00
Rastringin	Mean	2.43E+01	Rastringin	Mean	2.35E-01
	S.E.	2.85E+01		S.E	2.99E-01
Ackley	Mean	1.69E-08	Ackley	Mean	6.81E-15
	S.E.	2.28E-08		S.E	1.70E-15
Rosenbrock	Mean	9.87E-13	Rosenbrock	Mean	0.00E+00
	S.E.	1.49E-12		S.E	0.00E+00

The parameters for the tests are the ones in Table 20 where the null hypothesis tells us that the average of the Fuzzy Differential Evolution algorithm is greater than or equal to the average of the Differential Evolution algorithm. On the other hand, the alternative hypothesis establishes that the average of the Fuzzy Differential Evolution algorithm is less than the average of differential evolution algorithm, with a region of rejection for all values less than - 1.96, using the values in Table 21 we calculate the Z value for each of the functions, which are shown in Table 22. This Table also shows that for all the functions we can observe that we reject the null hypothesis, since the samples provide us with sufficient statistical evidence to support the alternative hypothesis.

Table 22. Results of applying the statistical z-test

Function	Original method	Proposed method	Z value	Evidence
Sphere	Differential Evolution	F. D. E. with Decreasing F	Z= -3.4344	Significant
Griewank	Differential Evolution	F. D.E. with Decreasing F	Z= -22.9992	Significant
Schwefel	Differential Evolution	F. D. E. with Decreasing F	Z= -3.4025	Significant
Rastringin	Differential Evolution	F. D. E. with Decreasing F	Z= -4.6200	Significant
Ackley	Differential Evolution	F.D. E. with Decreasing F	Z= -4.0661	Significant
Rosenbrock	Differential Evolution	F. D. E. with Decreasing F	Z= -3.6235	Significant

Summarizing the proposed fuzzy differential approach is significantly better than the original Differential Evolution method in all the Benchmark functions.

9. WILCOXON TEST STATISTICS

We decided to check our proposed algorithm of Fuzzy Differential Evolution (FDE) with two other fuzzy algorithms, and for this we use the Fuzzy Harmony Search algorithm (FHS) [14] and the Fuzzy Bat Algorithm (FBA) [15] since these two algorithms are using fuzzy logic as well for dynamic parameter adaptation as our proposed algorithm.

We considered the experiments with our of Fuzzy Differential Evolution algorithm (FDE) where F decreases because it is the way in which better results obtained with the set of Benchmark functions previously used. Table 23 shows the new set of functions used.

Table 23. Benchmark Functions

Function	Search Domain	$f$min
Sphere	$-5.12 \leq x_i \leq 5.12$	0
Rosenbrock	$-5 \leq x_i \leq 10$	0
Ackley	$-15 \leq x_i \leq 30$	0
Rastrigin	$-5.12 \leq x_i \leq 5.12$	0
}Zakharov	$-5 \leq x_i \leq 10$	0
Sum Squared	$-10 \leq x_i \leq 10$	0

In Figure 11 we can find the set of Benchmark functions listed in Table 23.

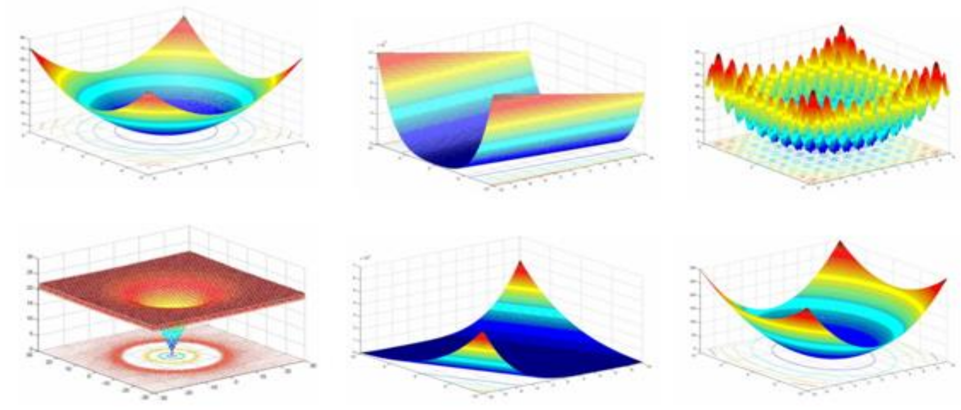

Fig.11. Benchmark Mathematical Functions.

Table 24 shows the parameters used for experiments, where F changes dynamically in decrease fashion; the search space used is that of each function listed in Table 23.

Table 24. Parameters of functions

Parameters
NP = 10,20,30,40,45 and 50
D = 10
CR = 0.1
GEN = 100

Experiments were carried out for different size of population, 30 experiments for population NP = 10, 30 for number of population NP = 20, up to the number of population of 50, later we obtained averages and we can observe these in Table 25.

Table 25. Averages by function

Average by function						
	N. of population					
Function	10	20	30	40	45	50
Sphere	1.81E-11	6.94E-43	1.97E-43	9.61E-22	9.60E-22	2.99E-43
Rosenbrock	4.08E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Ackley	7.98E-04	4.44E-15	4.20E-15	4.20E-15	4.20E-15	4.32E-15
Rastrigin	1.39E+00	3.32E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Zakharov	2.44E-12	4.41E-58	1.63E-59	3.86E-61	3.31E-61	1.39E-61
Sum Square	1.03E-06	4.34E-22	2.19E-21	1.17E-21	9.60E-22	1.29E-21

Taking into account the previous experiments we performed statistical testing of Wilcoxon, the first test we perform is with Fuzzy Bat Algorithm, the statistical test used for comparison is the Wilcoxon matched pairs test for analyzing the data, whose parameters are given in Table 26.

Table 26. Parameters for the statistical test

		F1	F2					
Function	No.	FBA	FDE	Difference	Abs(Difference)	Rank	Sign	Signed Rank
Spherical	1	3.97E-02	0.00E+00	3.97E-02	3.97E-02	1	1	1
Rosenbrock	2	6.85E-01	6.79E-02	6.17E-01	6.17E-01	6	1	6
Rastrigin	3	3.68E-01	2.38E-01	1.30E-01	1.30E-01	2	1	2
Ackley	4	3.66E-01	1.33E-04	3.65E-01	3.65E-01	4	1	4
Zakharov	5	3.32E-01	0.00E+00	3.32E-01	3.32E-01	3	1	3
Sum Square	6	4.40E-01	0.00E+00	4.40E-01	4.40E-01	5	1	5

The alternative hypothesis states that the average of the results of the Fuzzy Differential Evolution algorithm is different than the average performance of the Fuzzy Bat Algorithm, and therefore the null hypothesis tells us that the average of the results of the Fuzzy Differential Evolution algorithm is equal to the average of the Fuzzy Bat algorithm.

To test the hypothesis, first, the absolute values $|Z_i| \ldots |Z_n|$ are sorted and assigned its range Rank, Sign column indicates that all values obtained are positive, the column signed rank indicates the order of these values from lowest to highest.
The formula for the statistical test is defined as:

$$W^{+} = \sum_{\approx i > 0} R_i \qquad (12)$$

That is, the sum of the ranges $R_i$corresponding to positive valuesZ_i.
The value of W^{+} is the sum of the positive ranks, the value W- is the sum of the negative ranks, W is the differences between two data samples, and W0 indicates the value of the table for a two-tailed test using 30 samples.
The test to evaluate is as follows:

If W ≤ W0,

Then reject Ho.
Table 31 shows a statistical test applied to the two fuzzy methods is shown. With a confidence level of 95% and a value of W= 0 and W0 = 1. So the statistical test results are that: for the Fuzzy harmony search, there is significant evidence to reject the null hypothesis and the alternative hypothesis is accepted mentioning that the average Fuzzy Differential Evolution is different than the average performance of the fuzzy bat algorithm.

Table 31. Values of the parameters for the statistical test

W^{-}	W^{+}	**W**	**Level Significance**	**m = Degrees of freedom**	**W0 = W α,m =**
0	21	0	0.05	6	1

The following comparison is with the Fuzzy Harmony Search algorithm (FHS), Table 32 shows the used parameters.

Table 32. Parameters for the statistical test

		F1	F2					
Function	No.	FHS	FDE	Difference	abs(Difference)	Rank	Sign	Signed Rank
Spherical	1	1.38E-05	0.00E+00	1.38E-05	1.38E-05	3	1	3
Rosenbrock	2	9.53E-06	6.79E-02	6.79E-02	6.79E-02	5	0	-5
Rastrigin	3	0.00E+00	2.38E-01	2.38E-01	2.38E-01	6	0	-6
Ackley	4	4.73E-05	1.33E-04	8.57E-05	8.57E-05	4	0	-4
Zakharov	5	1.08E-08	0.00E+00	1.08E-08	1.08E-08	1	1	1
Sum Square	6	2.53E-06	0.00E+00	2.36E-06	2.36E-06	2	1	2

The alternative hypothesis states that the average of the results of the Fuzzy Differential Evolution algorithm is different than the average performance of the Fuzzy Harmony Search algorithm, and therefore the null hypothesis tells us that the average of the results of the Fuzzy Differential Evolution algorithm is equal to the average of the Fuzzy Harmony Search algorithm.
The formula of the statistical test that was applied is number 12.
The value of W^{+} is the sum of the positive ranks, the value W- is the sum of the negative ranks, W is the differences between two data samples, and W0 indicates the value of the table for a two-tailed test using 30 samples.

The test to evaluate is as follows:

If W $\leq$ W0, then fails to reject Ho.

Table 33 shows a statistical test applied to the two fuzzy methods is shown. With a confidence level of 95% and a value of W= 0 and W0 = 1.

Table 33. Values of parameters for the statistical test

W^{-}	W^{+}	**W**	**Level Significance**	**m = Degrees of freedom**	**W0 = W α,m =**
15	6	6	0.05	6	1

So the statistical test results are that:

There is not enough evidence to reject the null hypothesis therefore cannot accept the alternative hypothesis, this means that the Fuzzy Differential Evolution algorithm and the Fuzzy Harmony Search algorithm are statically the same.

10. CONCLUSIONS

We can conclude that setting dynamically the parameters of an evolutionary optimization method (in this case the Differential Evolution algorithm) can improve the quality of the results. In this work we are using fuzzy logic to dynamically change the F and Cr parameters of the algorithm. We have made several modifications to the algorithm and observed that the best results were obtained with modified F in decrement method and statistical evidence supports the conclusion to reject the null hypothesis in the comparison of the original algorithm against the proposed.
One of the main goals was to make the F and Cr parameters change dynamically in the Fuzzy Differential Evolution algorithm, but this combination of dynamic variables that not always the results we expected, the reason why we believe that this combination does not work is because the Cr variable is selected almost at random without taking into account that we can lose a good result, so do experiments by varying only Cr were not very good.

However using the Fuzzy Differential Evolution algorithm changing F in decrease we can observe that when performing the statistical test of Wilcoxon with other two fuzzy algorithms the proposed algorithm is competitive, although we are still working on the way in which our proposed algorithm better.
We can conclude that with only the modification of F change dynamically in the algorithm provides good results, in a matter of generations that the proposed algorithm produces better results in few generations to the original differential evolution algorithm, with at the runtime of the algorithm proposed by us is better. In general we can state that the proposed method was what we expected, we have achieved good results.

REFERENCES

[1] Aguas-Marmolejo S. J., Castillo O.: Optimization of Membership Functions for Type-1 and Type 2 Fuzzy Controllers of an Autonomous Mobile Robot Using PSO. Recent Advances on Hybrid Intelligent Systems 2013: 97-104

[2] Astudillo L., Melin P., Castillo O.: Optimization of a Fuzzy Tracking Controller for an Autonomous Mobile Robot under Perturbed Torques by Means of a Chemical Optimization Paradigm. Recent Advances on Hybrid Intelligent Systems 2013: 3-20

[3] Eftekhari M., Katebi S.D., Karimi M., A.H. Jahanmir: Eliciting transparent fuzzy model using differential evolution, School of Engineering, Shiraz University, Shiraz, Iran, Applied Soft Computing 8 (2008) 466–476.

[4] Fierro R., Castillo O., Design of Fuzzy Control Systems with Different PSO Variants. Recent Advances on Hybrid Intelligent Systems 2013: 81-88

[5] Hachicha N., Jarboui B., Siarry P.: A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics, InstitutSupérieur de Commerce et de Comptabilité de Bizerte, Zarzouna 7021, Bizerte, Tunisia, Information Sciences 181 (2011) 79–91

[6] Lizárraga E., Castillo O., Soria J.: A Method to Solve the Traveling Salesman Problem Using Ant Colony Optimization Variants with Ant Set Partitioning. Recent Advances on Hybrid Intelligent Systems 2013: 237-2461

[7] Melendez A., Castillo O.: Evolutionary Optimization of the Fuzzy Integrator in a Navigation System for a Mobile Robot. Recent Advances on Hybrid Intelligent Systems 2013: 21-31

[8]Melin P., Olivas F., Castillo O., Valdez F., Soria J.,GarcíaJ.: Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. ExpertSyst. Appl. 40(8): 3196-3206 (2013)

[9] Mezura-Montes E. andPalomeque-Ortiz A., Self-adaptiveandDeterministicParameter Control in DifferentialEvolutionforConstrainedOptimization.Efren Mezura-Montes, LaboratorioNacional de InformaticaAvanzada (LANIA A.C.), Rebsamen 80, Centro, Xalapa, Veracruz, 91000, MEXICO 2009.

[10] Neyoy H., Castillo O., José Soria: Dynamic Fuzzy Logic Parameter Tuning for ACO and Its Application in TSP Problems. Recent Advances on Hybrid Intelligent Systems 2013: 259-271.

[11] Oh S.-K., Kim W.-D., Pedrycz W., Design of optimized cascade fuzzy controller based on differential evolution: Simulation studies and practical insights, Department of Electrical Engineering, The University of Suwon, Engineering Applications of Artificial Intelligence 25 (2012) 520–532.

[12] Olivas F., Castillo O.: Particle Swarm Optimization with Dynamic Parameter Adaptation Using Fuzzy Logic for Benchmark Mathematical Functions. Recent Advances on Hybrid Intelligent Systems 2013: 247-258

[13] Peraza C., Valdez F. and Castillo O., A Harmony Search Algorithm Comparison with Genetic Algorithms, , Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics, Studies in Computational Intelligence 574,Springer International Publishing Switzerland 2015.

[14] Peraza Cinthia, Valdez Fevrier, Castillo Oscar, An Improved Harmony Search Algorithm Using Fuzzy Logic for the Optimization of Mathematical Functions, Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization,605-615,Springer International Publishing (2015).

[15] Perez Jonathan J., Valdez F., Castillo O. : A new bat algorithm with fuzzy logic for dynamical parameter adaptation and its applicability to fuzzy control design, Fuzzy logic augmentation of nature-inspired optimization meta heuristics, pp. 65-80, Springer (2015).

[16] Price, Storn R., Lampinen J. A., Differential Evolution, Kenneth V., Springer 2005.

[17] Price K., Storn R. and Lampinen J. A. Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), 2005: Springer

[18] Rahnamayan S., Tizhoosh, H. R., Salama M. M. A. Opposition-Based Differential Evolution, Evolutionary Computation, IEEE Transaction on (Volume: 12, Issue: 1), 2008, pp. 64-79.

[19] Raofen W., Zhang J., Zhang Y., Wang X.: Assessment of human operator functional state using a novel differential evolution optimization based adaptive fuzzy model, Lab for Brain-Computer Interfaces and Control, East China University of Science and Technology, Shanghai 200237, PR China, Biomedical Signal Processing and Control 7 (2012) 490– 498.

[20] Shi Y.-J., Teng H.-F., and Li Z.-Q., "Cooperative Co-evolutionary differential evolution for function optimization", Proc. 1st Int. Conf. Advances in Natural Comput, pp.1080 -1088, 2005

[21] Sk. I., Swagatam D., Saurav G., Subhrajit R., Ponnuthurai N. S., An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization, IEEE Transactions on systems, man, and Cybernetics—Part b: Cybernetics, vol. 42, no. 2, April 2012

[22] Sombra A., Valdez F., Melin P., Castillo O.: A new gravitational search algorithm using fuzzy logic to parameter adaptation. IEEE Congress on Evolutionary Computation 2013: 1068-1074

[23] Storn R. "On the usage of differential evolution for function optimization", Proc. Biennial Conf. North Amer. Fuzzy Inf. Process. Soc., pp.519 -523 1996

[24] Storn R. and Price K. Journal of Global Optimization 11, pp.341 -359 1997

[25] Swagatam D., Ajith A., Uday K. C., Amit K., Differential Evolution Using a Neighborhood-based Mutation Operator, Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India

[26] Sun J., Zhang Q. and Tsang E., "DE/EDA: A new evolutionary algorithm for global optimization", *Inf. Sci.*, vol. 169, pp.249 -262 2004.

[27] Valdez F., Melin P., Castillo O., Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making, in: Proceedings of the IEEE International Conference on Fuzzy Systems, 2009, pp. 2114–2119.

[28] Valdez F., Melin P., Castillo O.: Parallel Particle Swarm Optimization with Parameters Adaptation Using Fuzzy Logic. MICAI (2) 2012: 374-385

[29] Valdez F., Melin P., Castillo O.: Bio-inspired Optimization Methods on Graphic Processing Unit for Minimization of Complex Mathematical Functions. Recent Advances on Hybrid Intelligent Systems 2013: 313-322.

[30] Valdez F., P. Melin, O. Castillo: An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms. Applied Soft Computing 11 (2011) 2625–2632

[31] Vucetic D., Fuzzy differential evolution algorithm, The University of Western Ontario, London, Ontario, Canada, 2012.

[32] Wilcoxon, F. (1945) "Individual Comparisons by Ranking Methods. Biometrics 1, 80-83.

[33] Zelinka I. Real-time deterministic chaos control by means of selected evolutionary techniques, Engineering Applications of Artificial Intelligence 22 (2009) 283–297.

TIJUANA INSTITUTE OF TECHNOLOGY, TIJUANA, MÉXICO

E-mail address, Oscar Castillo: ocastillo@tectijuana.mx