A fuzzy logic approach for dynamic adaptation of parameters in galactic swarm optimization

Emer Bernal, Oscar Castillo, Jose Soria
Tijuana Institute of Technology
Tijuana, Mexico
emer_recargado@hotmail.com, ocastillo@tectijuana.mx, jsoria57@gmail.com

***Abstract*—In this article we propose the use of fuzzy systems for dynamic adjustment of parameters in the galactic swarm optimization (GSO) method. This algorithm is inspired by the movement of stars, galaxies and superclusters of galaxies under the force of gravity. GSO uses various cycles of exploration and exploitation phases to achieve a trade-off between the exploration of new solutions and exploitation of existing solutions. In this paper we proposed distinct fuzzy systems for the dynamic adaptation of the c_3 and c_4 parameters to measure the performance of the algorithm with 17 benchmark functions with different number of dimensions. In this paper a comparison was made between different variants to prove the efficacy of the method in optimization problems.**

Keywords—Galactic swarm optimization, GSO, fuzzy system, dynamic adaptation, benchmark function.

I. INTRODUCTION

We propose the galactic swarm optimization with dynamic adjustment of parameters utilizing fuzzy logic. We have designed different fuzzy systems for the c_3 and c_4 parameters to measure the performance of the fuzzy galactic swarm optimization against the original algorithm.

Zadeh originally proposed fuzzy logic and fuzzy set theory. Fuzzy systems are built based on rules formed through knowledge and heuristics based on human knowledge [10].

Today, fuzzy logic is considered as an effective technique for data management rule-based systems because of their tolerance for imprecision, ambiguity and lack of information. In addition systems based on rules allow knowledge representation and capturing personal experience into a set of if-then rules. In the rules a set of premises for achieving the conclusion of a result are evaluated [2].

Swarm intelligence techniques have gained popularity in recent decades because of their ability to find a relatively optimal solution for complex combinatorial optimization problems. The swarm intelligence techniques are approximate algorithms that incorporate a wide range of intelligent strategies inspired largely on natural processes, such as Particle swarm optimization (PSO), Grey wolf optimizer (GWO), Gravitational search algorithm (GSA) and Artificial Honey Bee Algorithm (AHB) [1], among others.

A new algorithm named galactic swarm optimization (GSO) was presented by Muthiah-Nakarajan and Noel. This paradigm increments the efficiency and effectiveness of the algorithm using various cycles of exploration and exploitation, thus increasing the possibilities of finding with precession the global minimum [14].

The GSO algorithm takes as inspiration the movement of the stars inside galaxies and movements of clusters and superclusters of galaxies. The population of potential solutions is divided into subpopulations, where all individual solutions are attracted towards the better solutions. The GSO algorithm is implemented by treating the best solution located by all individual subpopulations as a superswarm [14].

Since the PSO algorithm has been employed as inspiration for illustrating the proposed method is called to as the galactic swarm optimization (GSO), however, the method can be extended to different population based metaheuristics.

We are basing our work on the galactic swarm optimization presented originally by Muthiah-Nakarajan and Noel [14]. As related work to the Galactic Swarm Optimization we can find: A new global optimization metaheuristic inspired by galactic motion [14]. We based the idea of dynamic adaptation of parameters in the published work by Melin et al. in Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic[11].

The rest of the document is organized as shown below: in Section II a description of the Galactic swarm optimization is presented. In Section III we show the benchmark functions. In Section IV the methodology of our proposal for the adjustment of parameters is presented. In Section V the comparison of results is described so we can to observe the GSO algorithm behavior when implementing the fuzzy systems. In Section VI the conclusions obtained after the study of the galactic swarm optimization using benchmark functions are presented.

II. GALACTIC SWARM OPTIMIZATION

The GSO algorithm mimics the motion of stars, galaxies and superclusters of galaxies in the cosmos. Stars are not distributed uniformly in the cosmos, but clustered into galaxies, which in turn are not uniformly distributed. On a large enough scale individual galaxies appear as point masses.

The attraction of a set of stars within the galaxies and a set of galaxies to a larger set of galaxies is emulated with the algorithm GSO using the PSO algorithm, initially all individuals or solutions from each of the subpopulations are influenced by the best solutions in each subpopulation. Then each subpopulation is represented by the best solution in each of the subpopulations and treated as superswarm. The superswarm is composed with the best solutions of each subpopulation. In such a way that all individuals o solutions will are attracted towards the global best.

The swarms within the GSO algorithm area set X of D−tuples containing elements ($X_j^{(i)} \in \mathcal{R}^D$) consists of M partitions, called subswarmsX_i, each of size N. The elements of X are initialized with random values that are within the search space$[x_{min}, x_{max}]^D$.X_i is a swarm of size N.

In the first level of GSO, PSO is executed separately for each of the subswarms, since the population o swarm X is divided into M subpopulations in this way PSO is executed by M times. All the sub-swarms X_i are associated with a global best$g^{(i)}$ and are being updated if any of the personal best $p_j^{(i)}$have a value less than $g^{(i)}$,$f\left(p_j^{(i)}\right) < f\ (g^{(i)})$.

The motion of each subswarm X_i is independent and does not influence subswarmX_j for $i \neq j$, This allows our search to be total and not affected by getting the maximum performance of this exploration ability of several subswarms, a galactic best is defined as g, which is updated when some of the global bestsg$^{(i)}$contains a better value of function,$f(g^{(i)}) < f\ (g)$.This algorithm conserve record of the best solution by updating g.

All subswarms search independently in their search space, the search starts by calculating the velocity and position of the particles. The equations for updating them are as follows:

$$v_j^{(i)} \leftarrow \omega_1 v^{(i)} + c_1 r_1 \left(p_j^{(i)} - x_j^{(i)}\right) + c_2 r_2 \left(g^{(i)} - x_j^{(i)}\right) \quad (1)$$

$$x_j^{(i)} \leftarrow x_j^{(i)} + v_j^{(i)} \quad (2)$$

Where the inertial weight ω_1, and the random numbers r_1 and r_2 are given by:

$$\omega_1 = 1 - \frac{k}{L_1 + 1} \quad (3)$$

$$r_1 = \cup\ (-1, 1) \quad (4)$$

K is an integer number representing the current iteration ranging from0 toL_1. Eq. (4) representsr_i which a number is randomly chosen from the range of 1 to 1. Global bests collaborate in the following stage to form the superclusters. A new superswarm Y is produced by collecting the global bests from subswarms X_i.

$$y^{(i)} \in Y: i = 1, 2, \dots, M. \quad (5)$$
$$y^{(i)} = g_{(i)}$$

The velocity $v^{(i)}$and position vectors $y^{(i)}$are obtained from the following expressions shown:

$$v^{(i)} \leftarrow \omega_2 v^{(i)} + c_3 r_3 \left(p^{(i)} - y^{(i)}\right) + c_4 r_4 \left(g - y^{(i)}\right) \quad (6)$$

$$y^{(i)} \leftarrow y^{(i)} + v^{(i)} \quad (7)$$

Where $p^{(i)}$ is the personal best associated with the vectory$^{(i)}$.The expressions that defineω_2, r_3 and r_4 are similar to Eqs. (3) and (4). At this level g serves as global best and is updated only if a better solution is found, since the superswarm focuses on the best of each subswarm in this way can improve the exploitation.

The location of the galactic best after the final epoch g and its fitness value f (g) are returned as positions of minimum and cost of minimum respectively by the algorithm. The pseudocode is describes in Fig.1:

```
Level 1 Initialization X(i)j, V(i)j, P(i)j, g(i) within [xmin, xmax]^D Randomly.
Level 2 Initialization V(i), P(i), g within [xmin, xmax]^D Randomly.

for EP ← 1 to EPmax
do  Begin PSO:Level 1
    for i←1 to M
    do  for k←0 to L1
        do  for j←1 to N
            do  v(i)j ← ω1v(i) + c1r1(p(i)j − x(i)j) + c2r2(g(i) − x(i)j)
                x(i)j ← x(i)j + v(i)j
                if f(x(i)j) < f(p(i)j) then  p(i)j ← x(i)j;
                                             if f(p(i)j) < f(g(i)) then  g(i) ← p(i)j;
                                                                         if f(g(i)) < f(g)
                                                                         then g ← g(i);
    Begin PSO: Level 2
    Initialization Swarm y(i) = g(i) : i = 1, 2, … , M;
    for k ← 0 to L2
    do  for i←1 to M
        do  v(i) ← ω2v(i) + c3r3(p(i) − y(i)) + c4r4(g − y(i));
            y(i) ← y(i) + v(i);
            if f(y(i)) < f(p(i)) then  p(i) ← y(i);
                                       if f(p(i)) < f(g)
                                       then g ← p(i);
Return g, f(g)
```

Fig. 1. GSO pseudocode.

III. BENCHMARK FUNCTIONS

This part shows the benchmark functions utilized to measure the performance of the GSO algorithm with fuzzy dynamic adjustment of parameters.

In the area of algorithms for optimization it is common to use mathematical functions to test new methods and these are also used in this work: that is a modification to an optimization metaheuristic inspired on from the movement of stars inside galaxies and movements of clusters and superclusters of galaxies, known as GSO, where the fuzzy dynamic adaptation of the c_3 and $c_4$parameters will be made. The mathematical functions are listed below [4, 13]. These benchmark mathematical functions are usually considered in many works proposing new metaheuristics.

$$\boldsymbol{f_1(x) = \sum_{1=1}^{n} x_i^2} \quad (8)$$
$$\boldsymbol{with\, x_j \in [-5.12, 5.12] and f(x^*) = 0}$$

$$\boldsymbol{f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|} \quad (9)$$
$$\boldsymbol{with\, x_j \in [-10, 10] and f(x^*) = 0}$$

$$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2 \quad (10)$$
$$with\, x_j \in [-100, 100] and f(x^*) = 0$$

$$f_4(x) = max_i\{|x_i|,, 1 \leq i \leq n\} \quad (11)$$
$$with\, x_j \in [-100, 100] and f(x^*) = 0$$

$$f_5(x) = \sum_{i=1}^{n-1} \left[100\, (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (12)$$
$$with\, x_j \in [-30, 30] and f(x^*) = 0$$

$$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2 \quad (13)$$
$$with\, x_j \in [-100, 100] and f(x^*) = 0$$

$$f_7(x) = \sum_{i=1}^{n} i\, x_i^4 + random[0,1] \quad (14)$$
$$with\, x_j \in [-1.28, 1.28] and f(x^*) = 0$$

$$f_8(x) = \sum_{i-1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right) \quad (15)$$
$$with\, x_j \in [-500, 500] and f(x^*) = -418.9829x5$$

$$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] \quad (16)$$
$$with\, x_j \in [-5.12, 5.12] and f(x^*) = 0$$

$$f_{10}(x) = -20exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - exp\left(\frac{1}{n} \sum_{i=1}^{n} cos(2\pi x_i) \right) + 20 + e \quad (17)$$
$$with\, x_j \in [-32, 32] and f(x^*) = 0$$

$$Rosenbrock(x) = \sum_{i=1}^{n-1} \left[100(x_i + x_i^2)^2 + (x_{i-1})^2 \right] \quad (18)$$
$$with\, x_j \in [-5, 10] and f(x^*) = 0$$

$$Sum\, Squares(x) = \sum_{i=1}^{n} i x_i^2 \quad (19)$$
$$with\, x_j \in [-10, 10] and f(x^*) = 0$$

$$Zakharov(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i \right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i \right)^4 \quad (20)$$
$$with\, x_j \in [-5, 10] and f(x^*) = 0$$

$$Shubert(x) = \left(\sum_{i=1}^{5} icos((i+1) + i) \right) \left(\sum_{i=1}^{5} icos((i+1)x_2 + i) \right) \quad (21)$$
$$with\, x_j \in [-10,10]\ and\ f(x^*) = -186.7309$$

$$Baele(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2 \quad (22)$$
$$with\, x_j \in [-4.5,4.5]\ and\ f(x^*) = 0$$

$$Baele(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2 \quad (23)$$
$$with\, x_j \in [-4.5,4.5]\ and\ f(x^*) = 0$$

$$Booth(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (24)$$
$$with\, x_j \in [-10,10]\ and\ f(x^*) = 0$$

$$Dixon - Price(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i\,(2x_i^2 - x_{i-1})^2 \quad (25)$$
$$with\, x_j \in [-10,10]\ and\ f(x^*) = 0$$

IV. METHODOLOGY FOR PARAMETER ADAPTATION

The Galactic swarm optimization (GSO) algorithm is a strong search strategy utilized for solving complex optimization problems. In this article a new algorithm named Fuzzy Galactic swarm optimization (FGSO) with dynamic adjustment of parameters for the optimization of benchmark functions is proposed.

The appropriate parameter values of the fuzzy systems support the method to find the best solutions both global and local respectively. The primary objective is to obtain the best or optimal values of the parameters to achieve the best performance of the GSO algorithm. The fuzzy systems dynamically adapt the c_3 and c_4 parameters.

The fuzzy systems are of the Mamdani type with the input defined as the iterations and with two outputs respectively, the first is with c_3 in increase and c_4 in decrease and the second variant is with c_3 in decrease and c_4 in increase, as shown in Fig.2. The c_3 and c_4 parameters are the acceleration constants that provide the direction of the best local and global solutions.

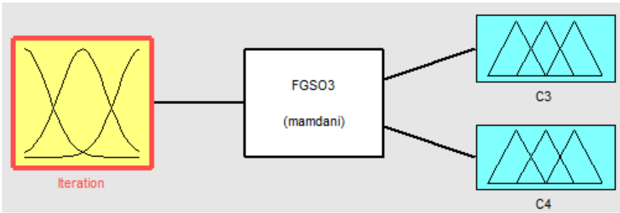

Fig. 2. Fuzzy system for the c_3 and c_4 parameters.

To measure the iterations of the algorithm, we utilize a percent of the iterations, that is, when the algorithm initiates the iterations will be considered "low", and as the iterations increases or approaches 100% is considered as "high". This notion is shown in the Eq. 26, as follows [11]:

$$Iterations = \frac{Current\ Iteration}{Maximum\ of\ Iterations} \qquad (26)$$

In Eq. 26, the current iteration indicates the iterations that have elapsed and the maximum of iterations indicates the total number of iterations for locate the best solution.

The input variable is shown in Fig. 3, which represents the iterations; the input variable was designed so that it was granulated in three triangular membership functions, Low, Medium and High.

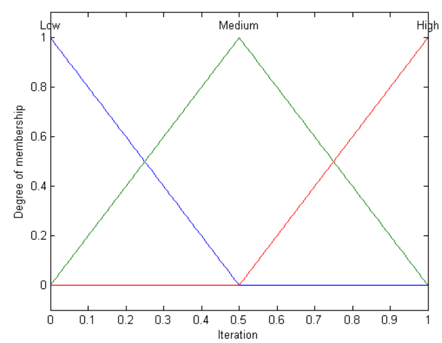

Fig. 3. Input Iteration.

The design of the output variables is illustrated in Fig. 4 and Fig. 5, which are representing c_3 and c_4 with a range from 0 to 3, and the outputs variables are granulated into three triangular membership functions, Low, Medium and High, respectively.

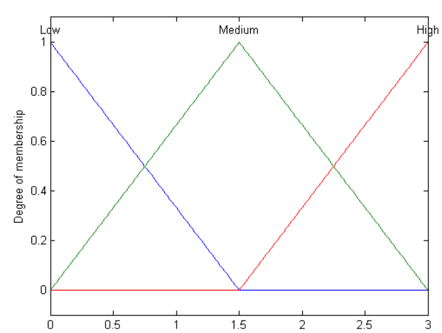

Fig. 4. Output c_3.

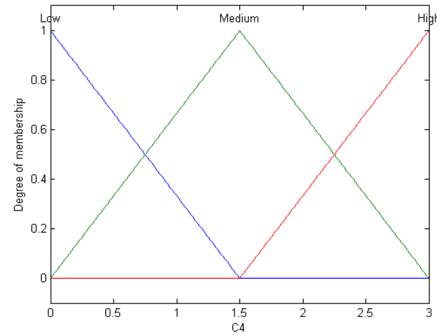

Fig. 4. Output c_4.

After designing the input and output variables of the systems, we design the fuzzy rules with the idea that they are on the increase and on decrease fashion respectively, as shown below.

The rules of the fuzzy system for c_3 on increase and c_4 on decrease:

1) If (Iteration is Low) then (c_3 is Low)(c_4 is High)
2) If (Iteration is Medium) then (c_3 is Medium)(c_4 is Medium)
3) If (Iteration is High) then (c_3 is High)(c_4 is Low)

The rules of the fuzzy system for c_3 on decrease and c_4 on increase:

1) If (Iteration is Low) then (c_3 is High)(c_4 is Low)
2) If (Iteration is Medium) then (c_3 is Medium)(c_4 is Medium)
3) f (Iteration is High) then (c_3 is Low)(c_4 is High)

V. COMPARISON OF RESULTS

In this section for the experiments with the galactic swarm optimization algorithm (GSO) we considered 17 benchmark functions with 10, 30 and 50 variables for the c_3 and $c_4$parameters. The results obtained by the GSO algorithm and our proposed method are shown in separate tables by the number of variables. Each table contains the average obtained after 30 executions of the algorithm.

Experiments with the galactic swarm optimization algorithm with fixed parameters as follows: the population is 5, the subpopulations are 10, iteration1 is 100, iteration2 is 1000 and epochs are 5 for 10 dimensions. The parameters for 30 dimensions are population is 5, subpopulations are 20, iteration1 is 150, iteration2 is 1500 and epochs are 5. Finally for 50 dimensions are population is 5, subpopulations are 20, iteration1 is 250, iteration2 is 1500 and epochs are 9. The parameters c_3 and c_4 have the following values $c_3$is 2 and c_4 is 2 for all cases.

The experiments for our propose method are the same as those of the original GSO algorithm only, but in this case c_3 and c_4 are dynamically adapted.

TABLE I. SIMULATIONS RESULTS FOR 10 DIMENSIONS

10 Dimensions			
Function	GSO	FGSO Inc-Dec	FGSO Dec-Inc
f1	0	0	0
f2	0	0	0
f3	0	0	0
f4	0	0	0
f5	0	5.7116E-06	5.4333E-06
f6	0	4.807E-06	4.922E-06
f7	0	1.35040296	1.34383565
f8	-4067.02898	-4102.30298	-4086.00795
f9	0	0	0
f10	0	0	0
Rosenbrock	6.6651E-07	7.5055E-12	1.3354E-09
Sumsquare	0	0	0
Zakharov	0	0	0
Shubert	-186.730909	-186.730909	-186.730909
Baele	3.0031E-21	1.9104E-24	1.4725E-23
Booth	0	0	0
Dixon-Price	0.07725316	0	0

Table I shows the results after the execution of 30 times the GSO algorithm and our proposal of dynamically adapting the c_3 and $c_4$parameters and we can observe the mean of the experiments for all the functions with 10 dimensions respectively.

TABLE II. SIMULATIONS RESULTS FOR 30 DIMENSIONS

30 Dimensions			
Function	GSO	FGSO Inc-Dec	FGSO Dec-Inc
f1	0	0	0
f2	0	0	0
f3	0	0	
f4	0	0	0
f5	0	3.9318E-07	1.9499E-07
f6	0	6.447E-07	9.6948E-07
f7	0	7.91095802	7.92191276
f8	-11366.0427	-11580.5481	-11460.0305
f9	0	0	0
f10	0	0	0
Rosenbrock	4.9109E-07	3.208E-08	5.1768E-08
Sum square	0	0	0
Zakharov	0	0	0
Shubert	-186.730909	-186.730909	-186.730909
Baele	0	0	0
Booth	0	0	0
Dixon-Price	0.66666668	0	0

Table IIshows the results after the execution of 30 times the GSO algorithm and our proposal of dynamically adapting the c_3 and $c_4$parameters and we can observe the mean of the experiments for all the functions with 30 dimensions respectively.

TABLE III. SIMULATIONS RESULTS FOR 50 DIMENSIONS

50 Dimensions			
Function	GSO	FGSO Inc-Dec	FGSO Dec-Inc
f1	0	0	0
f2	0	0	0
f3	0	0	0
f4	0	0	0
f5	0	8.1506E-10	5.2066E-11
f6	0	1.506E-09	2.2425E-09
f7	0	15.5608184	15.6088685
f8	-18626.6112	-18964.1188	-19211.8487
f9	0	0	0
f10	0	0	0
Rosenbrock	4.4041E-09	3.6926E-12	1.3196E-15
Sum square	0	0	0
Zakharov	0	0	0
Shubert	-186.730909	-186.730909	-186.730909
Baele	0	0	0
Booth	0	0	0
Dixon-Price	0.66666667	0	0

Table III shows the results after the execution of 30 times the GSO algorithm and our proposal of dynamically adapting the c_3 and $c_4$parameters and we can observe the mean of the experiments for all the functions with 50 dimensions respectively. For comparison purposes we perform a statistical test using the Wilcoxon-test [18], between our proposed method and the GSO algorithm. Tables IV and V show the means in pairs from the experiments used to analyze the data with 10 dimensions. The alternative hypothesis states that the mean of the experiments of the fuzzy galactic swarm optimization algorithm is different than the mean performance of the GSO algorithm, and therefore the null hypothesis tells us that the mean of the results of the fuzzy galactic swarm optimization algorithm is equal to the mean of the GSO algorithm.

TABLE IV. PARAMETERS FOR THE STASTICAL TEST BETWEEN FGSO INC-DEC AND GSO.

Function	No	GSO	FGSO Inc - Dec	Difference	Rank
f1	1	0	0	0	1.5
f2	2	0	0	0	1.5
f3	3	0	0	0	1.5
f4	4	0	0	0	1.5
f5	5	0	5.71162E-06	-5.71E-06	-13
f6	6	0	4.80701E-06	-4.81E-06	-14
f7	7	0	1.350402964	-1.35E+00	-16
f8	8	-4067.02897	-4102.30297	3.53E+01	17
f9	9	0	0	0	1.5
f10	10	0	0	0	1.5
Rosenbrock	11	6.66508E-07	7.50554E-12	6.67E-07	12
Sumsquare	12	0	0	0	1.5
Zakharov	13	0	0	0	1.5
Shubert	14	-186.730988	-186.730908	0	1.5
Baele	15	3.00306E-21	1.91041E-24	3.00E-21	11
Booth	16	0	0	0	1.5
Dixon-Price	17	0.077253156	0	7.73E-02	15

TABLE V. PARAMETERS FOR THE STASTICAL TEST BETWEEN FGSO DEC-INC AND GSO.

Function	No	GSO	FGSO Dec - Inc	Difference	Rank
f1	1	0	0	0	1.5
f2	2	0	0	0	1.5
f3	3	0	0	0	1.5
f4	4	0	0	0	1.5
f5	5	0	5.43329E-06	-5.43E-06	-13
f6	6	0	4.92204E-06	-4.92E-06	-14
f7	7	0	1.343835645	-1.34E+00	-16
f8	8	-4067.02897	-4086.00795	1.90E+01	17
f9	9	0	0	0	1.5
f10	10	0	0	0	1.5
Rosenbrock	11	6.66508E-07	1.3354E-09	6.65E-07	12
Sumsquare	12	0	0	0	1.5
Zakharov	13	0	0	0	1.5
Shubert	14	-186.730908	-186.730908	-6.25E-13	-11
Baele	15	3.00306E-21	1.47249E-23	2.99E-21	10
Booth	16	0	0	0	1.5
Dixon-Price	17	0.077253156	0	7.73E-02	15

To test the hypothesis, first, the values $|Z_i| \dots |Z_n|$ are sorted and assigned its corresponding Rank, the order of the values is from lowest to highest.

The expression for the statistical test was applied:

$$\boldsymbol{W_+} = \sum_{\boldsymbol{Z_i > 0}} \boldsymbol{R_i} \tag{27}$$

That is, the sum of the ranges $\boldsymbol{R_i}$ corresponding to positive values$\boldsymbol{Z_i}$.

The value of W+ is the sum of the positive ranks, the value W-is the sum of the negative ranks, W is the difference between both samples, and W0 represents the value of the table for two-tailed tests utilizing 30 samples as shown in Tables VI and VII.

The test to evaluate is as follows: If W $\leqslant$ W_0, Then fail to reject Ho.

TABLE VI. VALUES OF THE PARAMETERS FOR THE STASTICAL TEST BETWEEN FGSO INC-DEC AND GSO.

GSO and FGSO Inc - Dec					
W-	W+	W	Level Significance	m=Degrees of freedom	W_0 = W α,m =
43	58	43	0.05	17	35

Table VI shows a statistical test applied to the GSO algorithm and FGSO with c_3 in increase and c_4 in decrease. With a confidence level of 95% and a value of W= 43 and W_0 = 35. So the statistical test results are that: for the FGSO algorithm with c_3 in increase and c_4 in decrease, then we fail to reject Ho the null hypothesis and the alternative hypothesis is rejected mentioning that the average performance of the GSO algorithm with c_3 in increase and c_4 in decrease is equal than the average performance of the GSO algorithm.

TABLE VII. VALUES OF THE PARAMETERS FOR THE STASTICAL TEST BETWEEN FGSO DEC-INC AND GSO.

GSO and FGSO Dec - Inc					
W-	W+	W	Level Significance	m=Degrees of freedom	W_0 = W α,m =
54	74.5	54	0.05	17	35

Table VII shows a statistical test applied to the GSO algorithm and FGSO with c_3 in decrease and c_4 in increase. With a confidence level of 95% and a value of W= 54 and W_0 = 35. So the statistical test results are that: for the FGSO algorithm with c_3 in decrease and c_4 in increase, then we fail to reject Ho the null hypothesis and the alternative hypothesis is rejected mentioning that the average performance of the GSO algorithm with c_3 in decrease and c_4 in increase is equal than the average performance of the GSO algorithm.

VI. CONCLUSIONS

In the analysis of the results of each of our proposals, we can conclude that we have made improvements to the galactic swarm optimization, since our proposals, on average, are able to improve in some cases the original algorithm but nevertheless statistically one cannot say that our proposal is better. In this case the proposed approach can overcome the original GSO algorithm in most cases is the proposal from c_3 in increase and c_4 in decrease.

In this article various fuzzy systems developed for the dynamic adaptation of the parameters in the galactic swarm optimization algorithm are presented. We perform a comparison between these methods and the result of the experiments shows that our proposal can achieve an improvement against the original algorithm in some cases where the original algorithm fails to reach zero, based on the results shown in Tables I, II and III.

For future work we think of adding different inputs to the algorithm, applying the GSO algorithm for different control problems, perform hybridization with other metaheuristic algorithm to improve the algorithm GSO and migrate the GSO algorithm some other metaheuristic based population.

ACKNOWLEDGMENT

We would like to show our gratitude to the CONACYT and to Tijuana Institute of Technology for the help and resources provided for the development of our research.

REFERENCES

[1] E. Bernal, O. Castillo and J. Soria, “Imperialist Competitive Algorithm Applied to the Optimization of Mathematical Functions: A Parameter Variation Study", In Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization, vol. 601, pp. 219-232, Springer International Publishing, 2015.

[2] J. Cepeda-Negrete and R. E. Sanchez-Yanez, “Automatic selection of color constancy algorithms for dark image enhancement by fuzzy rule-based reasoning”. Applied Soft Computing, vol. 28, pp. 1-10, 2015.

[3] M. Y. Cheng and D. Prayogo, “Symbiotic Organisms Search: A new metaheuristic optimization algorithm”, Computers & Structures, vol. 139, pp. 98-112, 2014.

[4] A. R. Hedar, “Test functions for unconstrained global optimization” [online], Egypt, Assiut University, Available: http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm

[5] J. S. R Jang, C.T. Sun and E. Mizutani, “Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence”, 1997.

[6] A. H. Kashan, “A new metaheuristic for optimization: Optics inspired optimization (OIO)”, Computers & Operations Research, vol. 55, pp. 99-125, 2015.

[7] A. Kaveh and N. Farhoudi, “A new optimization method: Dolphin echolocation”, Advances in Engineering Software, vol. 59, pp. 53-70, 2013.

[8] B. S. Khehra, A. P. S. Pharwaha and M. Kaushal, “Fuzzy 2-partition entropy threshold selection based on Big Bang–Big Crunch Optimization algorithm”. Egyptian Informatics Journal, vol. 16(1), pp. 133-150, 2015.

[9] H. C. Kuo and C. H. Lin, “Cultural evolution algorithm for global optimizations and its applications”, Journal of applied research and technology, vol. 11(4), pp. 510-522, 2013.

[10] M. J. Mahmoodabadi and H. Jahanshahi, “Multi-objective optimized fuzzy-PID controllers for fourth order nonlinear systems”, Engineering Science and Technology, an International Journal, vol. 18, pp. 1084–1098, 2016.

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

[12] A. Milajić, D. Beljaković, N. Davidović, N. Vatin and V. Murgul, “Using the Big Bang–Big Crunch Algorithm for Rational Design of an Energy-Plus Building”, Procedia Engineering, vol. 117, pp. 916-923, 2015.

[13] S. Mirjalili, S.M Mirjalili and A. Lewis, “Grey wolf optimizer”, Advances in Engineering Software, vol. 69, pp. 46-61, 2014.

[14] V. Muthiah-Nakarajan and M.M. Noel, “Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion”, Applied Soft Computing, vol. 38, pp. 771-787, 2016.

[15] R. Oftadeh, M. J. Mahjoob and M. Shariatpanahi, “A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search”, Computers & Mathematics with Applications, vol. 60(7), pp. 2087-2098, 2010.

[16] M. Sedighizadeh and R. Bakhtiary, “Optimal multi-objective reconfiguration and capacitor placement of distribution systems with the Hybrid Big Bang–Big Crunch algorithm in the fuzzy framework”, Ain Shams Engineering Journal, vol. 7, pp. 113–129, 2016.

[17] F. Valdez, P. Melin and O. Castillo, “An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms”, Applied Soft Computing, vol. 11(2), pp. 2625-2632, 2011.

[18] F. Wilcoxon, “Individual Comparisons by Ranking Methods”, International Biometric Society, pp. 80-83, 1945.