Optimization of the Fuzzy C-Means Algorithm using Evolutionary Methods

Oscar Castillo, Elid Rubio, Jose Soria, and Enrique Naredo

Abstract—**This paper presents the optimization of the Fuzzy C-Means algorithm by evolutionary or bio-inspired methods, this in order to automatically find the optimal number of clusters and the weight exponent. Optimization methods used to realization of this paper were genetic algorithms and particle swarm optimization. The results obtained by both methods are presented, and a comparison between both methods to observe if one method is better than the other.**

***Index Terms*—Cluster validity, clustering number, comparison between methods, genetic algorithms, optimization, and particle swarm optimization.**

I. Introduction

Clusters of data arise from the need to find interesting patterns or groups of data with similar characteristics within a given data set. Fuzzy clustering aims at partitioning a data set into homogeneous fuzzy clusters. The most widely used algorithm to realize fuzzy clustering is the Fuzzy C-Means (FCM) algorithm proposed by Bezdek (1981) [1]. This algorithm has been the base to developing other clustering algorithms.

Although the fuzzy c-means algorithm is good in data clustering it has the inconvenient that finding the optimal number of clusters within a dataset is difficult, and the number of clusters has to be set arbitrarily, i.e. the number of clusters to be created by the clustering algorithm must be set manually on each algorithm execution, this is done again and again until finding the optimal number of clusters. Other factor that influences the performance of fuzzy c-means algorithm is the parameter m that is a weight exponent in the fuzzy membership, this parameter is normally m = 2 and works to find the optimal clusters number in some datasets but in other datasets not, which mean to each dataset the weight exponent is different.

Because of this, it is necessary to validate each of the fuzzy c-partitions once they are found, with different number of clusters and see which number of c-partitions is the optimal for a particular dataset. This evaluation process is called clustering validation. Currently there are many methods that have been proposed for the evaluation of fuzzy partitions, some of the methods of cluster validation which have been used in different works are: Partition Coefficient, Partition Entropy, Xie-Benis's Index, among others, mentioned in [1][2][3][4][5][6].

Due to that in the clustering algorithms is needed to predefine the number of clusters, and weight exponent m = 2, is not optimal for any dataset, and due to the importance that acquired the optimization, with evolutionary methods. In this research performed the optimization of fuzzy c-means algorithm, in order to find the optimal number of clusters and weight exponent to different datasets of automatic way.

Evolutionary methods used for optimization of the fuzzy c-means algorithm are genetic algorithms (GA) [7][8] and particle swarm optimization (PSO) [9][10], these evolutionary methods of optimization are used to find the optimal number of clusters and the weight exponent for different synthetics datasets.

II. Fuzzy C-Means algorithm

The Fuzzy C-Means algorithm is a clustering unsupervised method widely used in different pattern recognition works; this algorithm makes soft partitions where a datum can belong to different clusters with a different membership degree to each cluster. This clustering method is an iterative algorithm which uses the necessary condition to achieve the minimization of the objective function J_m represented by the following equation [1][3][4]:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \parallel x_j - v_i \parallel^2,\ m > 1 \tag{1}$$

Where n is the total number of patterns in a given data set and c is the number of clusters, which can be found from 2ton-1, $X = \{x_1, x_2, ..., x_n\} \subset R^s$ and $V = \{v_1, v_2, ..., v_n\} \subset R^s$ respectively are data characteristics and the centers of the clusters, and $U = [u_{ij}]_{c\times n}$ is a fuzzy partition matrix, which contains the membership degree of each dataset X to each cluster V. $\|x_j - v_j\|^2$ is the Euclidean distance between each data x_j of the dataset and the centers v_j of clusters, m is the weighting exponent which can influence the performance of the Fuzzy C-Means algorithm.

The corresponding centers of the clusters and membership degree to each respective data to solve the optimization problem with the constraints in (1) are given by equations (2)

Manuscript received September 9, 2011. This work was supported in part by the CONACYT and DGEST.

O. Castillo is with the Tijuana Institute Technology, Tijuana, Mexico (corresponding author phone: 664-623-6318; fax: 664-623-6318; e-mail: ocastillo@tectijuana.mx).

E. Rubio is a student with Tijuana Institute Technology, Tijuana, Mexico. (e-mail: elid@tectijuana.mx).

J. Soria is with Tijuana Institute Technology, Tijuana, Mexico (e-mail: jsoria57@gmail.com).

E. Naredo is a student with Tijuana Institute Technology, Tijuana, Mexico. (e-mail: enrique@tectijuana.mx).

(Advance online publication: 27 February 2012)

and (3) which provide an iterative procedure. The aim is to improve a sequence of fuzzy clusters until no further improvement in $J_m(U, V)$ can be performed [1][3][4]:

$$v_i = \frac{\sum_{j=1}^{n}(\mu_{ij})^m x_j}{\sum_{j=1}^{n}(\mu_{ij})^m}, \qquad 1 \leq i \leq c. \tag{2}$$

$$\mu_{ij} = \left[\sum_{k=1}^{c}\left(\frac{\|x_j - v_i\|^2}{\|x_j - v_k\|^2}\right)^{2/(m-1)}\right]^{-1}, \ 1 \leq i \leq c, 1 \leq j \leq n. \tag{3}$$

The Fuzzy C-Means algorithm consists of the following steps [3][5]:

1. Given a pre-selected number of clusters c and a chosen value for m, initialize the fuzzy partition matrix u_{ij} of x_j belonging to cluster I such that:

$$\sum_{i=1}^{c} \mu_{ij} = 1, \tag{4}$$

2. Calculate the center of the fuzzy cluster, v_j for i=1, 2,..., c using equation (2).
3. Use equation (3) to update the fuzzy membership u_{ij}.
4. If the improvement in $J_m(U, V)$ is less than a certain threshold(ε), then stop, otherwisegotostep2.

III. Cluster Validation

One of the main topics in data clustering is to evaluate the result of clustering algorithms. The problem is called cluster validation. More precisely, the cluster validation problem is to find an objective criterion to determine how good a partition generated by a clustering algorithm is. Since most clustering algorithms require a pre-assumed number of clusters, a validation criterion to find an optimal number of clusters would be very beneficial. Exist different validation index such as Partition Entropy, Partition Coefficient, Xie-Beni's index among other mentioned in [1][2][3][6].

We present our validation index for the Fuzzy C-Means algorithm. The index consists of two terms, the first term is a modification of the partition entropy index (13), this modification consist in squaring the first term to make a distinguishable variation of data between fuzzy partitions, figure 1 shows the behavior of the modified partition entropy, and figure 2 shows the behavior of partition entropy index; for a synthetic dataset with 2 dimensions and 2 clusters to find, from 2 to c numbers of clusters

$$I_{MPE} = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} {\mu_{ij}}^2 \log_2 \mu_{ij} \tag{13}$$

The second term is the sum of distances between the means of the fuzzy partitions (14); this measures the separation between fuzzy partitions of the fuzzy partitions matrix. The lower the value of the sum of the distances, the more separated fuzzy partitions of the partition matrix are. Figure 2 shows the behavior of the separation term on a synthetic dataset with 2 dimensions and 2 clusters to find, from 2 to c number of clusters.

$$D_{M_k} = \sum_{\substack{i,j=1 \\ i\neq j}}^{k} \| M_i - M_j \|^2, k = 1, \dots, c \tag{14}$$

Where M_k is the mean of the fuzzy partitions generated by the Fuzzy C-Means algorithm, which is defined by the following equation

$$M_k = \frac{\sum_{i=1}^{k}\mu_{ij}}{n}, k = 1, \dots, c, 1 \leq j \leq n \tag{15}$$

Where n is the total number of data into the dataset. The index proposes the addition of the results of equations (13) and (14). The proposed validation index is defined by the following equation:

$$I_{MPE-DMFP} = I_{MPE} + D_M \tag{16}$$

In general, we can define an optimal number of clusters $c*$ for the solution $min_{2\leq c\leq n-1}\, I_{MPE\text{-}DMFP}$ to produce a better performance by grouping the dataset X. Fig. 1 shows the behavior of the proposed validation index for a synthetic dataset with 2 dimensions and 2 clusters to finds, from 2 to c numbers of clusters.

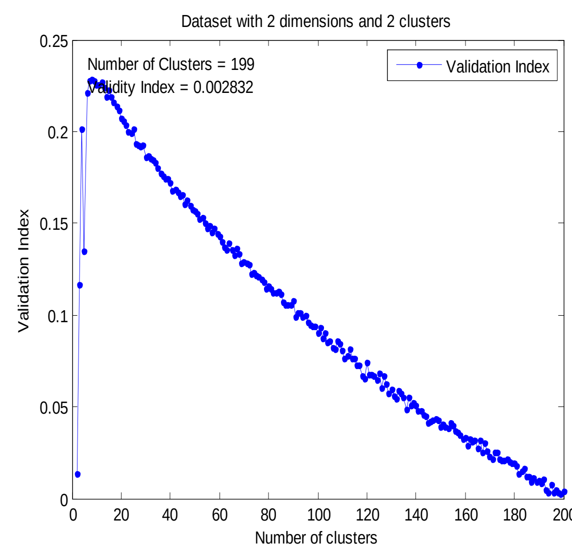

Fig. 1. Behavior of the modified partition entropy index

Fig. 1, shows the behavior of the modified partition entropy index, for this case the number of clusters found by the Fuzzy C-Means algorithm is 199 clusters with an index = 0.002832, This is because the closer the number of clusters to the number of data set, the smaller the value of modified partition entropy index is.

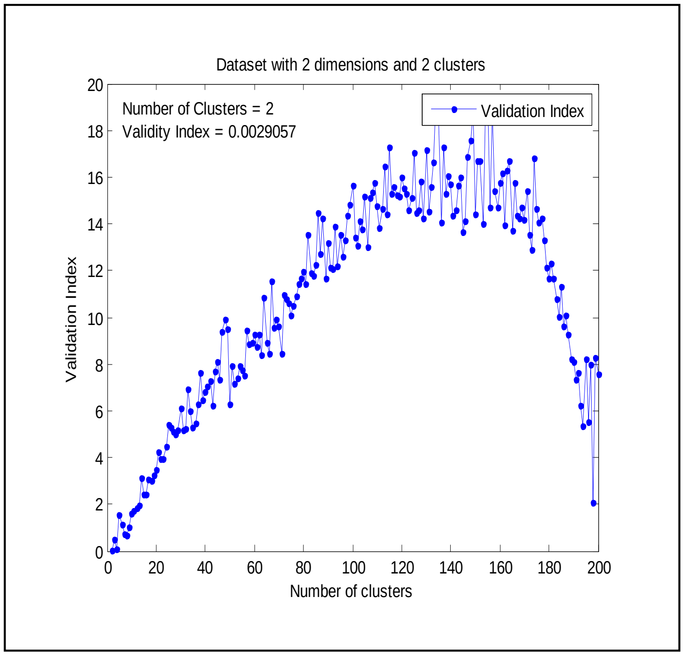

Fig. 2. Behavior of separation, based in the sum of distances between means of the fuzzy partitions of the matrix fuzzy partition

Figure 2, shows the behavior of separation and we can notice than the number of clusters is the correct one, which is 2 with an index = 0.0029057, and tell us that the sum of distances between means of fuzzy partitions is a validation index. This measure does not always finds the number of clusters because at times it met fuzzy partitions that are not well separated, but may improve the index of modified partition entropy to find the optimal number of clusters, keeping the maximum number of clusters for a data set is the one that gets the lowest validation index.

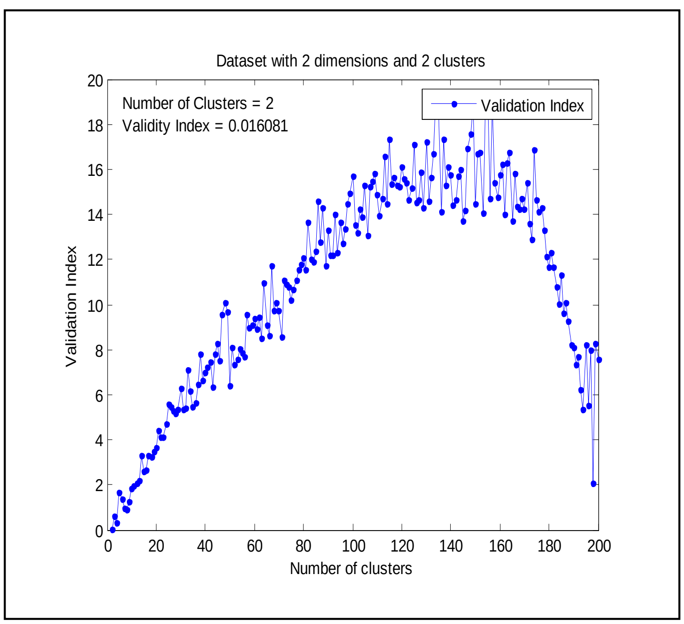

Fig. 3. Behavior of the proposed index.

In Fig. 3, we can see the behavior of the proposed index, and we can appreciate that the number of clusters is 2 which is correct with an index =0.016081, to avoid that the number of clusters is more closely to number of data, which is the lowest index number validation.

IV. Optimization of Fuzzy C-Means Algorithm

Optimization of Fuzzy C-Means algorithm is performed in order to find clustering number and weight exponent optimal, this due that these Fuzzy C-Means parameters are predefined to execution of algorithm.

Purpose of optimization Fuzzy C-Means algorithm is find the clustering number and the weight optimal of automatic way. To achieve this objective we used the optimization methods genetic algorithms (GA) and particle swarm optimization (PSO). Below show the methodology used for optimization of Fuzzy C-Means algorithm with optimization methods mentioned previously.

A. *Optimization of Fuzzy C-Means algorithm with GA*

Performance optimization using genetic algorithms is given by a sequence of steps, which are [7][8][11][12]:

1. Generate initial population.
2. Evaluate population
3. Selection.
4. Crossover.
5. Mutation.
6. Reinsertion of new individuals to the population.

From step 2 to step 6, it performs an iterative process until a stopping criterion is met, in Fig. 4 we can see the Scheme of GA for optimization of the Fuzzy C-Means algorithm (FCM).

In this figure we can observe that population evaluation is done by FCM algorithm, but for us to know how good some individuals need something that does not indicate the fitness of these, to measure aptitude of individuals evaluated by FCM, we use the proposed validation index mentioned in section III.

Individuals evaluated by the FCM algorithm, are formed only by two parameters as shown in Fig. 5, which are the number of clusters and the exponent of weight.

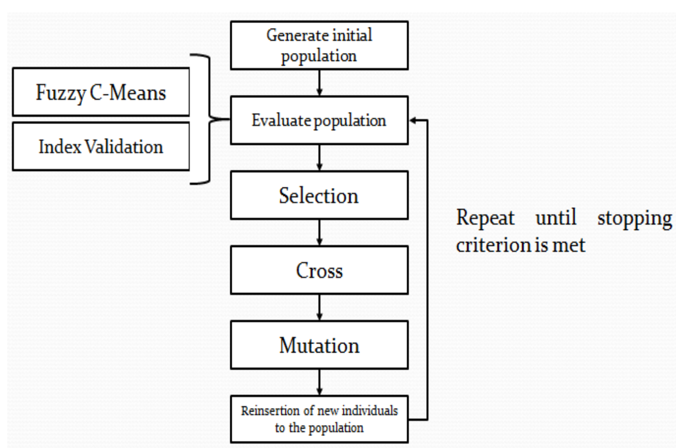

Fig. 4. Scheme of GA to optimization of Fuzzy C-Means algorithm

Individual (chromosome)	
Gen 1	Gen 2
Clustering Number	Weight Exponent

Fig. 5. Representation of an individual of the pobalcion.

Tests performed to optimization of the FMC algorithm with GAs, were done with synthetic data with dimensions from 2 to 8 and from 2 to 8 clusters for each number of dimensions, giving a total of 49 synthetic data sets, the results obtained for each tested dataset are shown in Table I. The parameters of the GA used to obtain these results are:

- Number of Individuals: 50.
- Number of generations: 25.
- Selection type: Stochastic Universal.
- Recombination Type: Discrete.
- Type of mutation: No Uniform.
- Selection rate: 0.90.
- Recombination rate: 0.90.
- Mutation rate: 0.10.
- Search space: Lower Limit: [2, 1.1], and High Limit: [√ (2 & n), 2.2], where n is the number of instances that make up data or data set.

B. Optimization of Fuzzy C-Means algorithm with PSO

The operation of the particles swarm optimization algorithm [9][10][13]-[18],is given by a sequence of steps, which are:

1. Generate initial swarm of particles.
2. Evaluating the particles swam.
3. Update particle velocity.
4. Calculate new positions of the particles.

From step 2to step 4, begins an iterative process until a stopping criterion is met, in Fig. 6 we can see the Scheme of the PSO for optimization of Fuzzy C-Means algorithm.

In this figure we can observe that particles' swarm evaluation is done by FCM algorithm, but for us to know how good some individuals need something that does not indicate the fitness of these, to measure aptitude of individuals evaluated by FCM, we use the proposed validation index mentioned in section III, the same way as in GA.

Particles evaluated by the FCM algorithm, are formed only by two parameters as shown in Fig. 7, which are the number of clusters and the exponent of weight.

Tests performed for the optimization of the FMC algorithm with PSO, were made with the same synthetic data sets used with GA, this in order to perform a fair comparison between the optimizations methods. The results obtained for each dataset tested with the PSO are shown in Table II. The parameters of the PSO used to obtain these results are:

- Number of Particles: 50.
- Number of Iterations: 25.
- Cognitive acceleration constant: 2.
- Social acceleration constant: 2.
- Constriction Factor: 1.
- Type of inertia: Decrease linear.

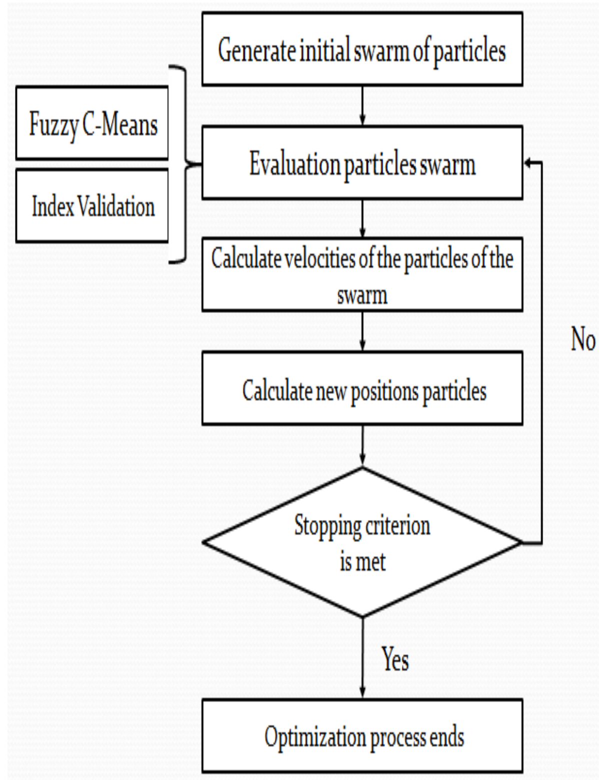

Fig. 6. Scheme of PSO to optimization of Fuzzy C-Means algorithm

Particle	
Variable 1	Variable 2
Clustering Number	Weight Exponent

Fig. 7. Representation of a particle swarm.

Tables of results obtained with both used methods of optimization, contain the following information:

- Mean and standard deviation of validation index.
- Mean and standard deviation of clustering number.
- Mean and standard deviation of weight exponent.
- Average time of execution.

The averages and standard deviations for the validation rate, the number of clusters, the exponent of fuzzification and the average execution times are obtained from 30 executions of the optimization methods. As we now from statistics, the averages and standard deviations of 30 tests are sufficient to allow using t student statistical tests, which may help us establish if there significant differences between the GA and PSO for the problem of optimizing the fuzzy clustering method. Tables I and II contain the results for each of the 49 synthetic data sets that were considered in the experiments.

Table I. Table of results obtained from FCM algorithm optimization with GA.

Dataset	Validation Index		Clustering Number		Weight Exponent		Average Time
	Mean	Std. Deviation	Mean	Std. Deviation	Mean	Std. Deviation	
Data2d2c	4.64E-13	2.05E-28	2.00	0.00	1.10	0.00	00:07.2seg.
Data3d2c	3.53E-13	1.54E-28	2.00	0.00	1.10	0.00	00:09.7seg.
Data4d2c	8.79E-12	4.93E-27	2.00	0.00	1.10	0.00	00:10.5seg.
Data5d2c	5.16E-14	3.85E-29	2.00	0.00	1.10	0.00	00:10.9seg.
Data6d2c	4.54E-14	1.28E-29	2.00	0.00	1.10	0.00	00:09.7seg.
Data7d2c	1.42E-15	3.95E-17	2.00	0.00	1.10	0.00	00:09.3seg.
Data8d2c	1.47E-13	2.57E-29	2.00	0.00	1.10	0.00	00:07.2seg.
Data2d3c	3.21E-02	3.45E-04	3.00	0.00	1.33	0.02	00:18.0seg.
Data3d3c	6.46E-10	3.53E-09	3.00	0.00	1.10	0.01	00:13.6seg.
Data4d3c	8.61E-16	6.02E-31	3.00	0.00	1.10	0.00	00:13.6seg.
Data5d3c	3.08E-08	1.69E-07	3.00	0.00	1.11	0.03	00:14.3seg.
Data6d3c	2.81E-15	4.01E-31	3.00	0.00	1.10	0.00	00:14.4seg.
Data7d3c	6.46E-11	1.31E-26	3.00	0.00	1.10	0.00	00:20.2seg.
Data8d3c	2.38E-10	2.10E-25	3.00	0.00	1.10	0.00	00:15.3seg.
Data2d4c	8.81E-08	6.73E-23	4.00	0.00	1.10	0.00	00:20.8seg.
Data3d4c	3.00E-06	1.52E-05	4.00	0.00	1.10	0.02	00:25.9seg.
Data4d4c	1.40E-03	5.48E-03	2.13	0.51	1.14	0.13	00:26.1seg.
Data5d4c	5.24E-08	2.87E-07	3.93	0.37	1.10	0.00	00:31.1seg.
Data6d4c	4.24E-15	3.60E-17	4.00	0.00	1.10	0.00	00:31.8seg.
Data7d4c	7.37E-10	4.04E-09	4.03	0.18	1.10	0.01	00:25.4seg.
Data8d4c	7.09E-09	3.88E-08	3.93	0.37	1.10	0.00	00:26.0seg.
Data2d5c	2.03E-01	7.11E-03	2.00	0.00	1.11	0.04	00:22.7seg.
Data3d5c	1.94E-02	3.10E-02	4.90	0.55	1.21	0.23	00:34.2seg.
Data4d5c	4.71E-02	7.41E-04	5.00	0.00	1.20	0.04	00:42.5seg.
Data5d5c	6.98E-03	3.42E-02	4.93	0.58	1.16	0.12	00:43.5seg.
Data6d5c	8.91E-03	3.49E-02	4.93	0.58	1.15	0.18	00:48.4seg.
Data7d5c	7.75E-14	5.37E-19	5.00	0.00	1.10	0.00	00:36.9seg.
Data8d5c	8.17E-06	4.47E-05	5.00	0.00	1.11	0.04	00:41.1seg.
Data2d6c	8.82E-02	8.13E-02	5.33	1.52	1.39	0.42	00:40.5seg.
Data3d6c	1.09E-02	1.09E-02	5.73	1.01	1.19	0.12	00:46.3seg.
Data4d6c	1.43E-03	2.85E-03	2.67	1.52	1.14	0.09	00:42.4seg.
Data5d6c	1.76E-06	5.67E-06	6.00	0.00	1.10	0.01	01:01.3seg.
Data6d6c	2.45E-03	1.34E-02	5.87	0.73	1.11	0.03	00:55.2seg.
Data7d6c	2.66E-04	1.46E-03	5.90	0.55	1.10	0.00	00:50.2seg.
Data8d6c	3.23E-08	1.77E-07	5.90	0.55	1.10	0.00	00:57.9seg.
Data2d7c	2.14E-02	3.56E-02	6.80	1.35	1.20	0.21	00:52.6seg.
Data3d7c	3.62E-02	2.78E-02	6.20	1.92	1.17	0.11	00:48.1seg.
Data4d7c	1.00E-02	3.62E-02	6.73	1.31	1.14	0.12	01:13.6min.
Data5d7c	4.12E-03	2.05E-02	6.87	0.94	1.12	0.06	01:25.1min.
Data6d7c	2.31E-03	1.23E-02	7.03	0.18	1.16	0.15	01:10.8min.
Data7d7c	4.88E-07	2.63E-06	7.03	0.18	1.11	0.05	01:02.7min.
Data8d7c	8.16E-03	3.11E-02	6.67	1.27	1.11	0.03	01:05.0min.
Data2d8c	8.25E-02	1.95E-02	3.43	1.28	1.36	0.08	00:56.4seg.
Data3d8c	1.65E-01	2.04E-02	5.03	3.09	1.68	0.18	00:52.3seg.
Data4d8c	4.37E-04	8.69E-04	7.40	1.57	1.14	0.08	01:22.6min.
Data5d8c	2.78E-02	7.28E-02	7.60	1.54	1.21	0.23	01:42.3min.
Data6d8c	4.22E-04	7.78E-04	6.60	2.58	1.11	0.02	01:24.3min.
Data7d8c	1.03E-08	4.03E-08	8.00	0.00	1.11	0.02	01:34.6min.
Data8d8c	8.91E-04	2.56E-03	7.30	1.95	1.11	0.03	01:33.3min.

Table II. Table of results obtained from FCM algorithm optimization with PSO.

Dataset	Validation Index		Clustering Number		Weight Exponent		Average Time
	Mean	Std. Deviation	Mean	Std. Deviation	Mean	Std. Deviation	
Data2d2c	4.64E-13	2.05E-28	2.00	0.00	1.10	0.00	00:04.1seg.
Data3d2c	3.53E-13	1.54E-28	2.00	0.00	1.10	0.00	00:05.7seg.
Data4d2c	8.79E-12	4.93E-27	2.00	0.00	1.10	0.00	00:06.7seg.
Data5d2c	5.16E-14	3.85E-29	2.00	0.00	1.10	0.00	00:05.9seg.
Data6d2c	4.54E-14	1.28E-29	2.00	0.00	1.10	0.00	00:11.7seg.
Data7d2c	1.56E-15	3.40E-17	2.00	0.00	1.10	0.00	00:05.7seg.
Data8d2c	1.47E-13	2.57E-29	2.00	0.00	1.10	0.00	00:05.2seg.
Data2d3c	3.20E-02	1.43E-06	3.00	0.00	1.32	0.00	00:16.1seg.
Data3d3c	7.06E-13	2.05E-28	3.00	0.00	1.10	0.00	00:12.0seg.
Data4d3c	8.61E-16	6.02E-31	3.00	0.00	1.10	0.00	00:10.5seg.
Data5d3c	3.97E-15	1.66E-17	3.00	0.00	1.10	0.00	00:12.2seg.
Data6d3c	2.81E-15	4.01E-31	2.97	0.18	1.10	0.18	00:14.5seg.
Data7d3c	6.46E-11	1.31E-26	3.00	0.00	1.10	0.00	00:17.1seg.
Data8d3c	2.38E-10	2.10E-25	3.00	0.00	1.10	0.00	00:15.0seg.
Data2d4c	8.81E-08	6.73E-23	4.00	0.00	1.10	0.00	00:20.3seg.
Data3d4c	2.16E-07	2.85E-19	4.00	0.00	1.10	0.00	00:23.5seg.
Data4d4c	3.74E-07	2.69E-22	2.00	0.00	1.10	0.00	00:13.7seg.
Data5d4c	1.05E-07	3.99E-07	3.87	0.51	1.10	0.51	00:27.1seg.
Data6d4c	4.25E-15	1.91E-17	4.00	0.00	1.10	0.00	00:31.3seg.
Data7d4c	6.66E-06	2.53E-05	3.87	0.51	1.10	0.51	00:21.2seg.
Data8d4c	1.42E-08	5.40E-08	3.90	0.55	1.10	0.55	00:33.6seg.
Data2d5c	2.01E-01	2.90E-10	2.00	0.00	1.10	0.00	00:12.1seg.
Data3d5c	7.25E-03	1.38E-06	5.00	0.00	1.10	0.00	00:34.1seg.
Data4d5c	6.13E-02	8.04E-02	4.90	0.55	1.24	0.55	00:38.2seg.
Data5d5c	6.25E-03	3.42E-02	4.90	0.55	1.10	0.55	00:37.5seg.
Data6d5c	1.04E-02	3.95E-02	4.80	0.76	1.11	0.76	00:41.9seg.
Data7d5c	7.75E-14	0.00E+00	5.00	0.00	1.10	0.00	00:44.3seg.
Data8d5c	6.55E-03	3.59E-02	4.83	0.59	1.10	0.59	00:47.0seg.
Data2d6c	9.92E-02	9.63E-02	4.87	1.78	1.39	1.78	00:24.4seg.
Data3d6c	6.63E-03	2.54E-03	4.93	1.80	1.13	1.80	00:26.9seg.
Data4d6c	1.37E-04	1.44E-12	2.00	0.00	1.10	0.00	00:22.1seg.
Data5d6c	5.80E-05	2.18E-04	5.73	1.01	1.10	1.01	00:46.1seg.
Data6d6c	2.68E-11	2.16E-19	6.00	0.00	1.10	0.00	00:50.6seg.
Data7d6c	3.32E-14	3.38E-20	6.00	0.00	1.10	0.00	01:01.0min.
Data8d6c	3.23E-08	1.77E-07	5.90	0.55	1.10	0.55	00:47.5min.
Data2d7c	1.46E-02	3.10E-02	6.67	1.27	1.13	1.27	00:55.8seg.
Data3d7c	3.95E-02	2.44E-02	5.03	2.53	1.20	2.53	00:56.8seg.
Data4d7c	1.43E-02	4.36E-02	6.53	1.55	1.10	1.55	01:00.4min.
Data5d7c	1.40E-02	3.61E-02	6.33	1.73	1.11	1.73	01:09.2min.
Data6d7c	1.51E-10	4.16E-19	7.00	0.00	1.10	0.00	01:22.5min.
Data7d7c	5.09E-16	8.93E-20	7.00	0.00	1.10	0.00	00:58.5seg.
Data8d7c	4.21E-03	2.30E-02	6.83	0.91	1.10	0.91	01:12.1min.
Data2d8c	8.10E-02	4.50E-02	2.93	0.25	1.39	0.25	00:25.5seg.
Data3d8c	1.71E-01	2.09E-02	3.80	2.80	1.71	2.80	00:35.0seg.
Data4d8c	1.07E-03	2.58E-03	7.03	2.14	1.10	2.14	00:32.2seg.
Data5d8c	2.33E-02	7.00E-02	7.20	1.92	1.10	1.92	00:32.1seg.
Data6d8c	6.03E-05	3.30E-04	7.80	1.10	1.10	1.10	01:41.1min.
Data7d8c	4.98E-05	1.90E-04	7.63	1.54	1.10	1.54	01:20.8min.
Data8d8c	2.80E-04	1.54E-03	7.80	1.10	1.10	1.10	01:29.3min.

(Advance online publication: 27 February 2012)

V. Comparison between optimizations Methods

This section presents a comparative study regarding the optimization methods used for the automation of FCM algorithm. Studies to compare the optimization methods used were based on the validation index, execution time. This comparison is because the results presented in the tables above, we note that the results are very similar, which is why the realization of the comparison.

To perform this comparison we used the results of both optimization methods to which is applied the T-Student test, which will tell us that based on the results of the sample of GA and to result sample of PSO optimization, if these methods are linearly separable, i.e. if there is a significant difference between the optimization methods (Tables III and IV).

Table III. Results of T-studentbased validation indices.

Dataset	GA		PSO		T-Student		
	Mean	Std. Deviation	Mean	Std. Deviation	T-Value	P-Value	Significant Diff.
Data2D2C	4.64E-13	1.03E-28	4.64E-13	1.03E-28	0.00	1.00	No
Data3D2C	3.53E-13	1.03E-28	3.53E-13	1.03E-28	0.00	1.00	No
Data4D2C	8.79E-12	3.29E-27	8.79E-12	3.29E-27	0.00	1.00	No
Data5D2C	5.16E-14	1.93E-29	5.16E-14	1.93E-29	0.00	1.00	No
Data6D2C	4.54E-14	0.00E+00	4.54E-14	0.00E+00	0.00	1.00	No
Data7D2C	**1.42E-15**	**3.95E-17**	**1.56E-15**	**3.40E-17**	**14.80**	**0.00**	**Yes**
Data8D2C	1.47E-13	0.00E+00	1.47E-13	0.00E+00	0.00	1.00	No
Data2D3C	3.21E-02	3.45E-04	3.20E-02	1.43E-06	1.80	0.08	No
Data3D3C	6.46E-10	3.53E-09	7.06E-13	4.11E-28	1.00	0.32	No
Data4D3C	8.61E-16	4.01E-31	8.61E-16	4.01E-31	0.00	1.00	No
Data5D3C	3.08E-08	1.69E-07	3.97E-15	1.66E-17	1.00	0.32	No
Data6D3C	2.81E-15	4.01E-31	2.81E-15	4.01E-31	1.00	0.32	No
Data7D3C	6.46E-11	0.00E+00	6.46E-11	0.00E+00	0.00	1.00	No
Data8D3C	2.38E-10	5.26E-26	2.38E-10	5.26E-26	1.00	0.32	No
Data2D4C	8.81E-08	5.38E-23	8.81E-08	5.38E-23	0.00	1.00	No
Data3D4C	3.00E-06	1.52E-05	2.16E-07	2.86E-19	1.00	0.32	No
Data4D4C	1.40E-03	5.48E-03	3.74E-07	2.15E-22	1.40	0.17	No
Data5D4C	5.24E-08	2.87E-07	1.05E-07	3.99E-07	0.58	0.56	No
Data6D4C	4.24E-15	3.60E-17	4.25E-15	1.91E-17	1.40	0.17	No
Data7D4C	7.37E-10	4.04E-09	6.66E-06	2.53E-05	1.44	0.16	No
Data8D4C	7.09E-09	3.88E-08	1.42E-08	5.40E-08	0.58	0.56	No
Data2D5C	2.03E-01	7.11E-03	2.01E-01	2.90E-10	1.14	0.26	No
Data3D5C	**1.94E-02**	**3.10E-02**	**7.25E-03**	**1.38E-06**	**2.16**	**0.04**	**Yes**
Data4D5C	4.71E-02	7.41E-04	6.13E-02	8.04E-02	0.97	0.34	No
Data5D5C	6.98E-03	3.42E-02	6.25E-03	3.42E-02	0.08	0.93	No
Data6D5C	8.91E-03	3.49E-02	1.04E-02	3.95E-02	0.15	0.88	No
Data7D5C	7.75E-14	5.37E-19	7.75E-14	0.00E+00	1.21	0.23	No
Data8D5C	8.17E-06	4.47E-05	6.55E-03	3.59E-02	1.00	0.32	No
Data2D6C	8.82E-02	8.13E-02	9.92E-02	9.63E-02	0.48	0.63	No
Data3D6C	**8.82E-02**	**1.09E-02**	**6.63E-03**	**2.54E-03**	**2.08**	**0.04**	**Yes**
Data4D6C	**8.82E-02**	**2.85E-03**	**1.37E-04**	**1.44E-12**	**2.49**	**0.02**	**Yes**
Data5D6C	8.82E-02	5.67E-06	5.80E-05	2.18E-04	1.41	0.16	No
Data6D6C	8.82E-02	1.34E-02	2.68E-11	2.16E-19	1.00	0.32	No
Data7D6C	8.82E-02	1.46E-03	3.32E-14	3.38E-20	1.00	0.32	No
Data8D6C	8.82E-02	1.77E-07	3.23E-08	1.77E-07	0.00	1.00	No
Data2D7C	2.14E-02	3.56E-02	1.46E-02	3.10E-02	0.79	0.43	No
Data3D7C	3.62E-02	2.78E-02	3.95E-02	2.44E-02	0.49	0.63	No
Data4D7C	1.00E-02	3.62E-02	1.43E-02	4.36E-02	0.42	0.68	No
Data5D7C	4.12E-03	2.05E-02	1.40E-02	3.61E-02	1.31	0.20	No
Data6D7C	2.31E-03	1.23E-02	1.51E-10	4.16E-19	1.03	0.31	No
Data7D7C	4.88E-07	2.63E-06	5.09E-16	8.93E-20	1.02	0.31	No
Data8D7C	8.16E-03	0.00E+00	4.21E-03	2.30E-02	0.56	0.58	No
Data2D8C	8.25E-02	1.95E-02	8.10E-02	4.50E-02	0.17	0.87	No
Data3D8C	1.65E-01	2.04E-02	1.71E-01	2.09E-02	1.12	0.27	No
Data4D8C	4.37E-04	8.69E-04	1.07E-03	2.58E-03	1.27	0.21	No
Data5D8C	2.78E-02	7.28E-02	2.33E-02	7.00E-02	0.24	0.81	No
Data6D8C	**4.22E-04**	**7.78E-04**	**6.03E-05**	**3.30E-04**	**2.34**	**0.02**	**Yes**
Data7D8C	1.03E-08	4.03E-08	4.98E-05	1.90E-04	1.44	0.16	No
Data8D8C	8.91E-04	2.56E-03	2.80E-04	1.54E-03	1.12	0.27	No

Table IV. Results of T-student based execution time.

Dataset	GA		PSO		T-Student		
	Mean	Std. Deviation	Mean	Std. Deviation	T-Value	P-Value	SignificantDiff.
Data2D2C	**00:07.2**	**00:01.0**	**00:04.6**	**00:00.4**	**13.42**	**0.00**	**Yes**
Data3D2C	**00:09.7**	**00:03.5**	**00:05.6**	**00:00.4**	**6.33**	**0.00**	**Yes**
Data4D2C	**00:10.5**	**00:01.4**	**00:06.4**	**00:00.5**	**15.65**	**0.00**	**Yes**
Data5D2C	**00:10.9**	**00:02.7**	**00:07.1**	**00:02.7**	**5.44**	**0.00**	**Yes**
Data6D2C	**00:09.7**	**00:01.7**	**00:07.7**	**00:03.7**	**2.70**	**0.01**	**Yes**
Data7D2C	**00:09.3**	**00:02.2**	**00:05.4**	**00:00.6**	**9.19**	**0.00**	**Yes**
Data8D2C	**00:07.2**	**00:01.0**	**00:05.3**	**00:00.7**	**9.16**	**0.00**	**Yes**
Data2D3C	**00:18.0**	**00:01.2**	**00:16.7**	**00:01.9**	**3.31**	**0.00**	**Yes**
Data3D3C	00:13.6	00:02.4	00:13.2	00:01.7	0.70	0.49	No
Data4D3C	**00:13.6**	**00:02.0**	**00:12.3**	**00:01.5**	**2.92**	**0.00**	**Yes**
Data5D3C	**00:14.3**	**00:01.5**	**00:13.4**	**00:02.0**	**2.10**	**0.04**	**Yes**
Data6D3C	00:14.4	00:01.8	00:15.3	00:02.2	0.70	0.49	No
Data7D3C	**00:20.2**	**00:02.2**	**00:18.1**	**00:02.7**	**3.38**	**0.00**	**Yes**
Data8D3C	00:15.3	00:01.8	00:14.0	00:01.9	0.70	0.49	No
Data2D4C	00:20.8	00:02.2	00:19.9	00:01.5	1.95	0.06	No
Data3D4C	00:25.9	00:02.3	00:25.6	00:01.9	0.44	0.66	No
Data4D4C	**00:26.1**	**00:07.6**	**00:13.2**	**00:01.1**	**9.21**	**0.00**	**Yes**
Data5D4C	**00:31.1**	**00:02.9**	**00:25.9**	**00:04.4**	**5.43**	**0.00**	**Yes**
Data6D4C	00:31.8	00:03.6	00:31.2	00:03.3	0.72	0.48	No
Data7D4C	00:25.4	00:04.1	00:24.2	00:03.7	1.23	0.22	No
Data8D4C	00:26.0	00:02.7	00:27.8	00:05.7	1.56	0.12	No
Data2D5C	**00:22.7**	**00:06.0**	**00:14.1**	**00:01.3**	**7.65**	**0.00**	**Yes**
Data3D5C	00:34.2	00:04.3	00:32.3	00:02.8	1.99	0.05	No
Data4D5C	**00:42.5**	**00:02.9**	**00:38.1**	**00:04.3**	**4.63**	**0.00**	**Yes**
Data5D5C	**00:43.5**	**00:05.2**	**00:35.8**	**00:04.7**	**6.10**	**0.00**	**Yes**
Data6D5C	**00:48.4**	**00:06.8**	**00:41.7**	**00:07.2**	**3.73**	**0.00**	**Yes**
Data7D5C	**00:36.9**	**00:02.6**	**00:41.7**	**00:02.7**	**6.89**	**0.00**	**Yes**
Data8D5C	00:41.1	00:03.3	00:40.7	00:06.1	0.37	0.71	No
Data2D6C	00:40.5	00:06.1	00:37.0	00:08.6	1.79	0.08	No
Data3D6C	**00:40.5**	**00:06.1**	**00:34.5**	**00:10.1**	**5.48**	**0.00**	**Yes**
Data4D6C	**00:40.5**	**00:17.0**	**00:20.3**	**00:02.5**	**7.06**	**0.00**	**Yes**
Data5D6C	**00:40.5**	**00:03.6**	**00:47.8**	**00:08.4**	**8.15**	**0.00**	**Yes**
Data6D6C	**00:40.5**	**00:04.0**	**00:51.3**	**00:05.1**	**3.28**	**0.00**	**Yes**
Data7D6C	00:40.5	00:05.1	00:48.7	00:05.0	1.13	0.26	No
Data8D6C	**00:40.5**	**00:07.0**	**00:49.9**	**00:06.7**	**4.55**	**0.00**	**Yes**
Data2D7C	00:52.6	00:09.7	00:49.0	00:08.6	1.52	0.13	No
Data3D7C	**00:48.1**	**00:08.1**	**00:41.6**	**00:11.8**	**2.51**	**0.02**	**Yes**
Data4D7C	**01:13.6**	**00:11.4**	**00:58.9**	**00:12.9**	**4.66**	**0.00**	**Yes**
Data5D7C	**01:25.1**	**00:12.9**	**01:08.2**	**00:16.7**	**4.39**	**0.00**	**Yes**
Data6D7C	**01:10.8**	**00:04.5**	**01:16.4**	**00:12.2**	**2.37**	**0.02**	**Yes**
Data7D7C	01:02.7	00:06.8	01:06.8	00:11.0	1.75	0.09	No
Data8D7C	01:05.0	00:01.0	01:06.9	00:10.9	0.85	0.40	No
Data2D8C	**00:56.4**	**00:10.5**	**00:43.1**	**00:05.7**	**6.13**	**0.00**	**Yes**
Data3D8C	00:52.3	00:14.1	00:49.0	00:22.9	0.67	0.51	No
Data4D8C	**01:22.6**	**00:18.6**	**01:11.2**	**00:17.2**	**2.46**	**0.02**	**Yes**
Data5D8C	01:42.3	00:16.9	01:41.1	00:28.5	0.19	0.85	No
Data6D8C	01:24.3	00:20.1	01:31.1	00:15.5	1.46	0.15	No
Data7D8C	01:34.6	00:05.8	01:35.3	00:19.7	0.18	0.86	No
Data8D8C	01:33.3	00:17.0	01:37.0	00:18.2	0.80	0.42	No

VI. Conclusion

In this research work was performed optimization of FCM algorithm, where by the optimization methods used are seeking to find the optimal number of clusters and the exponent of fuzzification.

In the presented results with different optimization methods, it was possible to observe that in most of the averages of groups of data sets, the average number of clusters is approximately the number of clusters and in some cases the group average is the number clusters containing the data set, showing that for some cases, both the GA and PSO are efficient for optimization of FCM algorithm.

Because it is not seen clearly significant differences between the optimization methods used in the presented results, we made a t-student test, this in order to know if there was a significant difference between the optimization methods used for optimization the FCM algorithm.

Where we can observe in the result in terms of validation index show in Table III, only 10% (5/49 datasets) of data sets used in which there is a significant difference and 90% (44/49 datasets) of sets data in which there is no significant difference, therefore, based on this statistical test we can say that both optimization methods are good, the optimization of the FCM algorithm.

In Table IV we can observe in the result in terms of execution time that 59% (29/49 datasets) of data sets used in which there is a significant difference and 49% (20/49 datasets) where no significant difference, based on this we can say that one method is better than another in terms of speed of execution, from our point of view PSO is faster than GA because PSO performs fewer operations than GA.

REFERENCES

[1] J. Yen; R. Langari; *"Fuzzy Logic: Intelligence, Control, and Information"*, Upper Saddle River, New Jersey; Prentice Hall, 1999.

[2] K. L. Wu, M. S. Yang; "A cluster validity index for fuzzy clustering", Pattern Recognition Letters, Volume 26, Issue 9, 1 July 2005, Pages 1275-1291.

[3] M. K. Pakhira, S. Bandyopadhyay, U. Maulik, "A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification", Fuzzy Sets and Systems, Volume 155, Issue 2, 16 October 2005, Pages 191-214.

[4] R. Kruse, C. Döring, M. J. Lesot; "Fundamentals of Fuzzy Clustering"; In: Advances in Fuzzy Clustering and its Applications; John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2007, Pages 3-30.

[5] W. Wang, Y. Zhang; "On fuzzy cluster validity indices", Fuzzy Sets and Systems, Volume 158, Issue 19, Theme: Data Analysis, 1 October 2007, Pages 2095-2117.

[6] Y. Zhang, W. Wang, X. Zhang, Y. Li; "A cluster validity index for fuzzy clustering", Information Sciences, Volume 178, Issue 4, 15 February 2008, Pages 1205-1218.

[7] J. H. Holland, *"Adaptation in Natural and Artificial Systems"*, 2a ed., MIT Press, 1992.

[8] D.Goldberg, *"Genetic Algorithms in Search, Optimization and Machine Learning"*, Addison Wesley, 1989.

[9] J. Kennedy, R. Eberhart, *"Particle Swam Optimization"*, in Proc. IEEE Int. Conf. Neural Network (ICNN), Nov. 1995, vol. 4, pages: 1942-1948.

[10] R. Eberhart, J. Kennedy, *"A new optimizer using particle swarm theory"*, in proc. 6th Int. Symp. Micro Machine and Human Science (MHS), Oct. 1995, pages: 39-43.

[11] K. F. Man, K. S. Tang, S. Kwong. "*Genetic Algorithms: Concepts and Designs",* Springer-Verlag, 1999.

[12] Randy L. Haupt and Sue Ellen Haupt, *"Practical Genetic Algorithms"*. John Wiley & Sons, Inc., 1998.

[13] Y.del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.-C.Hernandez, Harley R.G., *"Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems"*, Evolutionary Computation, IEEE Transactions on, Apr 2008, pages: 171-195.

[14] R. Eberhart, Y. Shi. J. Kennedy, *"Swam Intelligence"*, San Mateo, California. Morgan Kaufmann, 2001.

[15] A. P. Engelbrecht, *"Fundamentals of Computational Swarm Intelligence"*, John Wiley & Sons, 2006.

[16] H. J. Escalante, M. Montes, L. E. Sucar, *"Particle Swarm Model Selection"*, Journal of Machine Learning Research 10, 2009, pages: 405-440.

[17] R. Eberhart, Y. Shi, *"Particle swarm optimization: Developments, applications and resources"*, in Proceedings of the IEEE Congress on Evolutionary Computation, May 2001, vol. 1, pages: 81–86.

[18] X. Hu, Shi Y., R. Eberhart, *"Recent advances in particle swarm"*, in Proceeding of the IEEE Congress on Evolutionary Computation, Jun 2004, vol. 1, pages: 90–97.