An Efficient Intelligent Clustering Tool based on Hybrid Fuzzified Algorithm for Electrical Data

A. Jaya Mabel Rani [1]
[1]Research Scholar,
Sathyabama University
Chennai, India

Latha Parthiban[2]
[2]*Department of CSE,*
Pondicherry University,
Pondicherry, India.

N.M. Jothi Swaroopan[3]
Department of EEE
RMK Engineering College
Chennai, India.

ABSTRACT

Fuzzified optimization based data clustering is one of the important data mining tool which is active research of real world problems. This paper proposed Fuzzified Particle Swarm Optimization and K-Harmonic Means algorithm (FPSO+KHM) for clustering the electrical data systems. The partitioned clustering algorithms are more suitable for clustering large datasets. The K-Harmonic means algorithm is center based clustering algorithm and very insensitive to the selection of initial partition using built in boost function, but easily trapped in global optima. The proposed algorithm uses Fuzzified PSO and K-Harmonic means to generate more accurate, robust , better clustering results. This algorithm can generate the solution in few number of iterations, and get faster convergence when compare to K-Harmonic Means and hybrid PSO+ K-Harmonic Means algorithms. This algorithm is applied for two different set of IEEE standard electrical bus data systems.

Keywords

Clustering, Fuzzified PSO, K-Harmonic means algorithm, Convergence, local optima, IEEE bus system.

1. INTRODUCTION

In Electrical Power systems, there is a specific and growing interest towards electrical bus data organization especially from the point of distribution side [1]. The emerging trend towards unbundling the distribution and supply services, as well as new degrees of freedom enabled by the legislation for location up dedicated tariff offers, specific customer group which provides new opportunities to the supply operators for introducing a advanced metering technologies by receiving several data from the field, optimally by quality and dedicated Clustering algorithms.

Clustering is a basic operation used in unsupervised document organization, automatic topic extraction [2], and information retrieval [3] etc., Clustering involves dividing a set of objects into a specified number of clusters [4, 5]. In recent years, partitioned clustering technique [6 - 8] is well suited for clustering a large document dataset due to their relatively low computational requirements. The time complexity of the partitioning technique is almost linear, which makes it widely used. The best-known partitioning clustering algorithm is the K-means algorithm and its variants

[9 - 11], because it can be easily implemented and is the most capable one in terms of the execution time. The major disadvantage of this algorithm is that it is sensitive to the selection of the initial partition and may converge to local optima. To overcome this difficulty used K-harmonic Means algorithm [12, 13] with built in boost function for initial cluster, but it easily convergence in local optima. To overcome this drawback used globalised Particle Swarm Optimization (PSO) algorithm [14]. This PSO algorithm can be used to generate good initial cluster centroids for the K-Harmonic Means. The popularity of Particle Swarm Optimization is due to its significant property of dealing with the optimization problems without any restrictions on the structure or type of the function to be optimized and due to the ease of computation. However, a critical problem in PSO is the large number of iterations or high computation time required to obtain the near global optimum. Thus, an amendment in classical PSO technique is incorporated using fuzzy logic strategy for obtaining a faster convergence and fuzzy logic have emerged [15] as a balancing artificial tool to conventional deterministic mathematical methods called Fuzzified Particle Swam Optimization (FPSO)

2. K-HARMONIC MEANS ALGORITHM

The K-Harmonic Means algorithm (KHM) is the improvement of K-means algorithm which uses the Harmonic Averages (HA) of the distances from each point to the centers. K-Harmonic means takes the sum over all data points of the harmonic function. Let *x* $=\{x_1, \dots \dots x_n\}$, the data to be clustered. *M*= $\{m_l | l = 1, \dots, K\}$ be K centers. Then the membership function $mem(m_l/x_i)$ is defined by the proposition of data point x_i that belongs to center m_l. In K-Harmonic Means, the quality of the result is measured by the sum of a function *d(x, M)* over all *x*, using equation (1).

$$f(\mathrm{X}, \mathrm{M}) = \sum_{x \epsilon X} d(x, M) \qquad (1)$$

The *d(x,M)* is calculated using equation (2)

$$d(x, M)\ HA\ \{\|x-m\|^2 / m \epsilon M\} = {}^{|M|}\Big/ \sum_{m \epsilon M} \frac{1}{\|x-m\|^2} \qquad (2)$$

The Harmonic average of K numbers $\{a_{1\,,} . . . . a_k\}$ is defined as equation (3).

$$HA(\{a_1\ ,\ . . . . , a_k\}) = {}^{K}\Big/ \sum_{k=1}^{k} \frac{1}{a_k} \qquad (3)$$

The harmonic average is small if one of the a_k is small. Therefore *HA()* behaves more like the *MIN()* function . This desired property need for defining performance fitness function for measuring clustering quality function. The K-Harmonic Means performance fitness function measured by the given equation (4).

$$f(X,M) = \sum_{i=1}^{n} HA\,\{\|x_i - m_l\|^2/l = 1\,, \dots.\,, K\}$$

$$= \textstyle\sum_{i=1}^{N} \frac{K}{\sum_{l=1}^{K}\frac{1}{\|x_{i-}m_l\|^2}} \quad (4)$$

Here *d(x,m)* =||*x-m*||^{2,} which does not have the desired weighting function in its recursive optimization. This algorithm used weighting function $w(x_i)$, that define how much influence data point x_i has in re-computing the center parameters in the next iteration. So the desired weighting function can be derived theoretically by using the *p*th power of the l^2-distance as *d(x,M)* in KHM . The algorithm for K-Harmonic means with weighted function is given below.

1. Randomly choose the initial centers.
2. Calculate performance fitness function using equation (4) with weighted function p, where p>2.
3. For each data pointx_i, compute its membership function $mem(m_l/x_i)$ in each center m_l according to the equation. (5).

$$mem(m_l/x_i) = \frac{\|x_i - m_l\|^{-p-2}}{\sum_{l=1}^{k}\|x_i - m_l\|^{-p-2}} \quad (5)$$

4. For each data pointx_i, compute its weight$w(x_i)$ according to the equation (6).

$$w(x_i) = \frac{\sum_{l=1}^{k}\ \|x_i - m_l\|^{-p-2}}{\left(\sum_{l=1}^{k}\ \|x_i - m_l\|^{-p}\right)^2} \quad (6)$$

5. For each center m_l ,recomputed its location from all data points x_i according to their membership and weights by using equation (7)

$$m_l = \frac{\sum_{i=1}^{n} mem(m_l/x_i)w(x_i)\mathrm{x_i}}{\sum_{i=1}^{n} mem(m_l/x_i)w(x_i)} \quad (7)$$

6. Repeat step 2 to 5 predefined number of iterations or until $d(x,M)$ change significantly.

7. Assign data point x_i to cluster *l* with the biggest $mem(m_l/x_i)$.

3. PSO BASED K-HARMONIC MEANS ALGORITHM

In recent clustering analysis, Particle Swarm Optimization (PSO) is being widely implemented due to its simplicity and efficiency [16] when, compared with the techniques like

means, Fuzzy *C*-means [17], *K*-Harmonic means and other traditional clustering approaches. In general, the PSO algorithm produces better results with reference to inter-cluster and intra-cluster distances [18], while having quantization errors comparable to the other algorithms. PSO was originally developed by Eberhart and Kennedy in 1995 [19, 20], and was inspired by the social behavior of a bird flock. In the PSO algorithm [21, 22] the birds in a flock are emblematically represented as particles. These particles can be considered as simple agents “flying” through a problem space. A particle’s location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a performance fitness function that provides a quantitative value of the solution’s utility. The velocity and location of each particle moving along each dimension of the problem space will be altered with each generation of movement. In combination, the particle’s personal experience, *pbest* and its neighbor’s experience, *gbest* influence the movement of each particle through a problem space. The random values, *rand$_1$* and *rand$_2$*, are used for the sake of completeness, that is, to make sure that particles explore wide search space before converging around the optimal solution. The values of c$_1$ and c$_2$ control the weight balance of *pbest* and *gbes*t in deciding the particle’s next movement velocity. For every generation, the particle’s new location is computed by adding the particle’s current velocity, V-vector, to its location, X-vector. Mathematically, given a multi-dimensional problem space, the *i*th particle changes its velocity and location according to the following equations (8, 9)

$$v_i^{t+1} = k\,(\omega * v_i^t + c_1 \,.\, rand_1(pbest_i^t - x_i^t) + c_2 . rand_2\,(gbest^t - x_i^t)) \quad (8)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (9)$$

Where *k* is the constriction factor [13].

$$k = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|} \quad (10)$$

where c$_1$ + c$_2$ = c and c > 4 .The acceleration constants c$_1$and c$_2$ pulls each particle toward *pbest* and *gbest* positions. x_i^t and v_i^t are the current position and velocity of the ith particle; *pbest* is the location of the particle that experiences the best fitness value; *gbest* is the location of the particles that experience a global best fitness value; *rand$_1$*, *rand$_2$* are random values in the range of (0, 1); *ω* denotes the inertia weight factor [23] is given in the following equation (11)

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} \times t \quad (11)$$

t_{max} is the maximum number of iterations or generations. ω_{max} and ω_{min} are the upper and lower limit of the inertia weight [24]. The inertia weight balances global and local explorations and it decreases linearly from 0.9 to 0.4 in each run. The value of inertia weight is computed based on the iteration (*t* & *tmax*) alone (independent of the problem being solved). Equation (8) requires each particle to record its current coordinate x_i^t its velocity v_i^t that indicates the speed of its movement along the dimensions in a problem space, and the coordinates *pbest* and *gbest* where the best performance fitness values were computed. The best performance fitness values are updated at each generation, based on equation (12),

$$pbest_i^{(t+1)} = \begin{cases} pbest_i^t & f\big(x_i(t+1)\big) \leq f\big(x_i(t)\big) \\ x_i(t+1) & f\big(x_i(t+1)\big) > f\big(x_i(t)\big) \end{cases} \quad (12)$$

Where the symbol f *d*enotes the performance fitness function in equation (4); *pbest* stands for the best fitness values and the coordination where the value was calculated; and *t* denotes the generation step. The basic PSO algorithm is given in the flow chart as shown in figure 1.

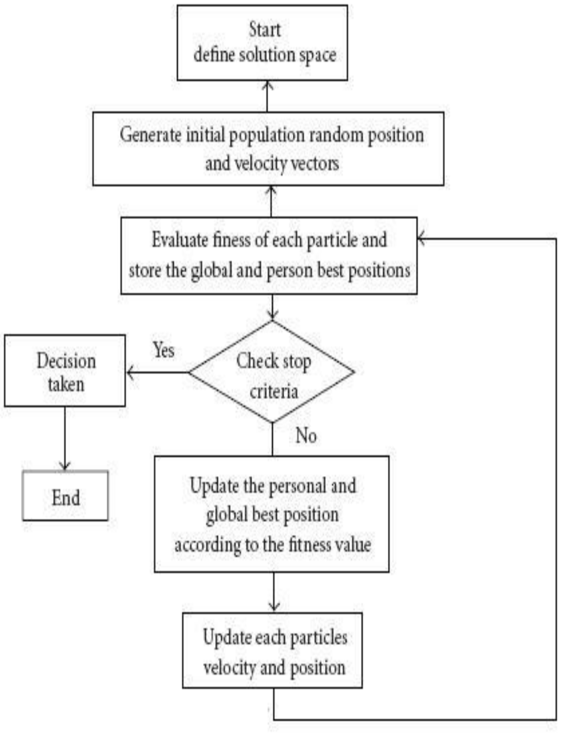
Start
define solution space
Generate initial population random position and velocity vectors
Evaluate finess of each particle and store the global and person best positions
Check stop criteria
Yes
No
Decision taken
End
Update the personal and global best position according to the fitness value
Update each particles velocity and position

Fig.1. The general flowchart of PSO

A single particle in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of candidate clustering solutions for the document collection. Each particle maintains a matrix $X_i = (m_1, m_2, \ldots, m_i, \ldots, m_k)$, where, m_i represents the ith cluster centroid vector and k is the cluster number. At each iteration, the particle adjusts the centroid vector position in the vector space according to its own experience and those of its neighbors. The harmonic average distance between a cluster centroid and a document is used as the performance fitness value to evaluate the solution represented by each particle. The performance fitness value is measured by the equation (13).

$$f(X,M) = \sum_{i=1}^{N} \frac{K}{\sum_{l=1}^{K} \frac{1}{\|x_{i-}m_l\|^2}} \quad (13)$$

The algorithm for PSO based K-Harmonic means is given below.

Step 1. Initialize the parameters for PSO and K-Harmonic means (KHM) with swarm size n, acceleration coefficient $c1$, $c2$, ω_{max}, and ω_{min}.

Step 2. Create a swarm with *P* particles .

Step 3. Initialize x_i, v_i, *pbest* for each particle and *gbest* for the swarm.

Step 4. PSO algorithm: (module 1)
- 4.1 Calculate the cluster centers for each particle.
- 4.2 Calculate the fitness value of each particle.
- 4.3 Calculate *pbest* for each particle.
- 4.4 Calculate *gbest* for the swarm.
- 4.5 Update the velocity and position for each particle.

Step 5. If the terminating condition of PSO is not met, go to step 4.

Step 6. K-Harmonic Means (KHM) algorithm. (Module 2)
- 6.1 Randomly choose the initial centers.
- 6.2 Calculate performance fitness function using equation (4) with weighted function p, where p = 3.
- 6.3 For each data pointx_i, compute its membership function $mem(m_l/x_i)$ in each center m_l according to the equation. (5).
- 6.4 For each data pointx_i, compute its weight $w(x_i)$ according to the equation (6).
- 6.5 For each center m_l ,recomputed its location from all data points x_i .

Step 7. If KHM terminating condition is not met, go to step 6.

Step 8. If PSO+KHM terminating condition is not met, go to step 4.

4. FUZZIFIED PSO

The proposed fuzzified PSO is one of the fastest growth algorithm for clustering of large dataset. Fuzzy logic provides a mathematical framework where the entire range of the data lies in between 0 and 1. Fuzzy based PSO [25] is easier to implement than evolutionary algorithms because it only involves a single operator for updating solutions.

In traditional PSO technique the value of inertia weight factor *ω* [26] is computed completely based on iteration count (t and t_{max}) and it is independent of the problem being solved which leads to slow and premature convergence[27]. Hence there is a need for an adaptive inertia weight. The convergence depends on the relative performance fitness function value as equation (14).

$$f = {f(X,M)} \Big/ {f(X,M)_{max}} \quad (14)$$

where $f(X,M)_{max}$ is the maximum fitness value in the present swarm. The relative performance fitness function value f is an essential factor which has a major influence in the convergence process. If the relative performance fitness value *f* is lower then the corresponding inertia weight should small and vice versa. The other factor that influences the convergence is the search range (x_j^{max} - x_j^{min}) which is a constant throughout the whole search process rather a self adaptive search range would be more appropriate. Thus these two factors need a certain control to obtain a better convergence. Moreover the relationship between them seems arbitrary, complex and ambiguous to determine, hence fuzzy logic strategy would be more appropriate than a crisp relation.

The various sequential steps involved in the FPSO based algorithm are same as that in section 3 except the calculation of inertia weight factor in the velocity updating process as follows,

(i) The fuzzy logic inputs and output are decided and their feasible ranges are declared. The two fuzzy inputs are as follows:

Input 1 = f based on equation (15)

$Input\ 2= Max \{(X_i^{\ max\ t} - X_i^{\ t\ i\ iter}) (X_i^{\ t\ i\ iter}\ X_i^{\ min\ t}) \}$ (16)

The *Input 1* is the first essential factor and *Input 2* is an active search range determined as the maximum search distance or range pertaining to each distance of the vector in the present iteration from any of its corresponding limits (maximum or minimum). The output of the fuzzy logic strategy is the inertia weight *ω*.

(ii) Fuzzification of inputs and output using triangular membership function. Five fuzzy linguistic sets have been used for each of the inputs and output as shown in Figure 2.

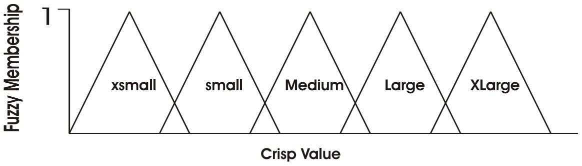

Fig. 2. Fuzzy Membership Function

(iii) The fuzzy rule base in Table 1 is formulated for all combinations of fuzzy inputs based on their ranges.
iv) Defuzzification of output using Centroid method.

$$C = \sum_{i=1}^{5} X_{\ i}\ Y_{\ i} \ /\ \sum_{i=1}^{5} Y_{\ i} \qquad (17)$$

here $X_{\ i}$ is the mid-point of each fuzzy output set and $Y_{\ i}$ is its corresponding membership function value. The centroid C is scaled (multiplied by range of inertia weight) ω_{max}-ω_{min} to obtain inertia weight value of each element in the particle.

Table 1. Fuzzy Rule Base

Input 1	**XSmall**	**Small**	**Medium**	**Large**	**XLarge**
Input 2					
XSmall	XSmall	XSmall	Small	Small	Small
Small	XSmall	Small	Small	Medium	Medium
Medium	XSmall	Small	Medium	Large	Large
Large	Small	Medium	Large	XLarge	XLarge
XLarge	Small	Medium	Large	XLarge	XLarge

5. FUZZIFIED PSO BASED K-HARMONIC MEANS ALGORITHM

In this proposed method PSO algorithm is based on fuzzification and defuzzification process and also called fuzzified PSO. All the sequential evaluations are same as section 4 except the fuzzification and defuzzification of PSO. Here fuzzified PSO is hybrid into K-Harmonic means algorithm and called as Fuzzzified PSO + KHM algorithm. Using this algorithm can avoid trapping of local optima and increase convergence speed. The existing Fuzzy PSO the acceleration coefficient c_1 and c_2 are adapted, but in this Fuzzied PSO adaptation based on Inertia weight (ω). This is more advanced than existing Fuzzy PSO. Because the iteration count based on inertia weight (ie, constant). When adapting inertia weight, iteration depends on size of the data. So it can avoid premature convergence or wastage of convergence. Here the Fuzzy logic gives the promising results if the data sets are too complex to analyze or available information is inexact or indecisive.

This fuzzified PSO base K-harmonic means algorithm also has two modules. 1. Fuzzified PSO module 2. K=Harmonic Means module. First apply this Fuzzified PSO module for the initialization of the data. Then this module is transferred to the K-harmonic means modules for the faster convergence. Usually K-harmonic means algorithms are faster than PSO algorithm . Because of both hybrid algorithm it can get advantage of fuzzy based PSO and K-harmonic means algorithm. The algorithm of this proposed method is same as PSO based K-Harmonic Means algorithm with the fuzzification of ω in PSO.

```
Step 1. Initialize the parameters for PSO and K-Harmonic means (KHM) with swarm size n, acceleration coefficient $c1$ , $c2$ , $\omega_{max}$, and $\omega_{min}$,and initial cluster.
Step 2. Create a swarm with P particles .
Step 3. Initialize $x_i$, $v_i$, pbest for each particle and gbest for the swarm.
Step 4. FPSO algorithm: (module 1)
    4.1 Calculate the cluster centers for each particle.
    4.2 Calculate the fitness value of each particle.
    4.3 Update fuzzy membership function as given in section 4 to calculate ω.
    4.4 Calculate pbest for each particle.
    4.5 Calculate gbest for the swarm.
    4.6 Update the velocity and position for each particle.
    4.7 If the terminating condition of FPSO is not met, go to step 4.
Step 5. K-Harmonic Means algorithm. (Module 2)
    5.1 Randomly choose the initial centers.
    5.2 Calculate performance fitness function using equation (4) with weighted function p, where p=3.
    5.3 For each data point$x_i$, compute its membership function $mem(m_l/x_i)$ in each center $m_l$ according to the equation. (5).
    5.4 For each data point$x_i$, compute its weight $w(x_i)$ according to the equation (6).
    5.5 For each center $m_l$ ,recomputed its location from all data points $x_i$ .

Step 6. If KHM terminating condition is not met, go to step5.
Step 7. If FPSO+KHM terminating condition is not met, go to step 4.
```

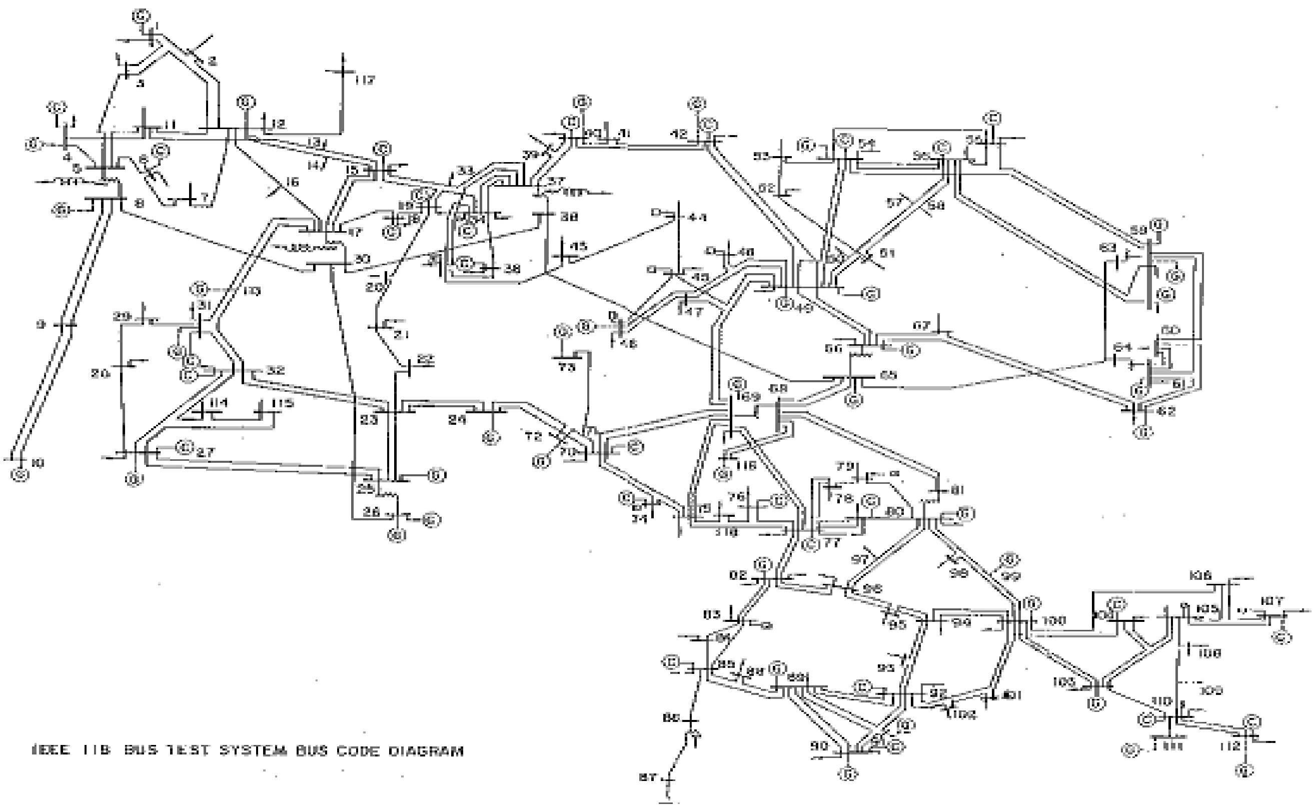

Fig.3. IEEE 118 bus line diagram

6. EXPERIMENTAL DATA SETS AND PROPOSED RESULTS

The proposed algorithm is tested on two different data sets of IEEE 300 bus and IEEE 118 electrical bus data system. The IEEE 118 bus line diagram is shown in figure 3 and the sample data set are given in table 2. The data set contains Voltage V, Active Power Generation Pg, Reactive Power Generation Qg, Active Power Demand Pd, Reactive Power Demand Qd, Shunt Conductance Gsh, and Shunt Susceptance Bsh. Sample descriptions of database are given in table 3. The experimental results are simulated using JAVA simulation on a Pentium 2.66 GHZ. The parameters used in the KHM, PSO and FPSO are as follows: $c_1 = c_2 = 2.05$, $\omega_{max} = 2.0$ and $\omega_{min} = 0.2$ The swarm size n for PSO and FPSO is 1000. The fuzzy logic data for inertia weight computation is given in table 4.

The quality of clustering is evaluated using KHM, PSO+KHM and FPSO+KHM for IEEE 300 Bus system are shown in Table 5. The results show superior performance in terms of less Harmonic mean, Standard deviation, Iterations and CPU time when compared to PSO+KHM and KHM.

Table 2. Sample Data for IEEE 118 Bus Systems

Type	V	Pg	Qg	Pd	Qd	Gsh	Bsh
PQ	1.00	0.00	0.00	0.70	0.230	0	0.00
PV	0.990	0.85	0.00	0.470	0.100	0	0.00
PQ	1.000	0.00	0.00	0.240	0.040	0	0.00
PV	0.984	0.00	0.00	0.590	0.260	0	0.14
PQ	1.000	0.00	0.00	0.200	0.110	0	0.15

Table 3. Sample Descriptions of data base

DATA SET	Number of Buses	Number of Data	Number of Classes
IEEE 300 bus system	300	2100	5
IEEE 180 bus system	180	1260	3

Table 4. Data for Fuzzy Inertia Weight

Fuzzy set	Input 1	Input 2*	Output
XSmall	0.00001to 0.00004	0.1% - 15%	0.2 to 0.6
Small	0.00003 to 0.006	13% - 50%	0.4 to 1.0
Medium	0.005 to 0.05	45% - 65%	0.8 to 1.4
Large	0.03 to 0.5	60% - 85%	1.2 to 1.8
XLarge	0.4 to 1	80% - 100%	1.6 to 2

(*Input 2 is expressed in percentage of control variable bound range {Upper limit – Lower limit})

Table 5. Performance comparisons of KHM, PSO+KHM and FPSO+ KHM Algorithms for IEEE 300 Bus

IEEE 300	KHM	PSO+KHM	FPSO+KHM
Harmonic Mean	757.811	756.617	756.253
Standard Deviation	0.028	0.079	0.007
No. of Iterations	140	100	60
CPU time(ms)	60	45	30

Table 6. Performance comparisons of KHM, PSO+KHM and FPSO+ KHM Algorithms for IEEE 118 Bus

IEEE 118	KHM	PSO+KHM	FPSO+KHM
Harmonic Mean	243.456	240.450	235.440
Standard Deviation	0.256	0.250	0.012
No. of Iterations	100	80	40
CPU time(ms)	50	30	20

Similarly the quality of clustering is evaluated using KHM, PSO+KHM and FPSO+KHM for IEEE 118 Bus system are shown in Table 6. The results show superior performance in terms of less Harmonic mean, Standard deviation, Iterations and CPU time when compared to PSO+KHM and KHM.

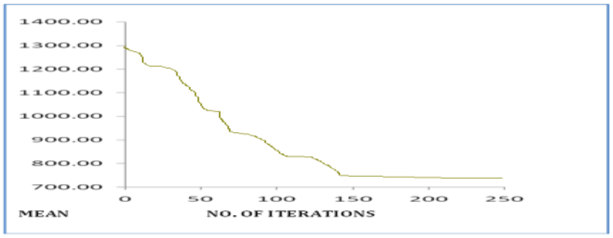

Figure. 4. Convergence characteristics of KHM

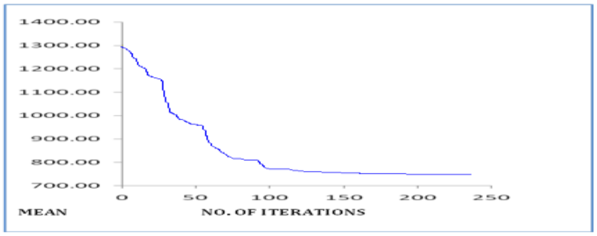

Figure. 5. Convergence characteristics of PSO+KHM

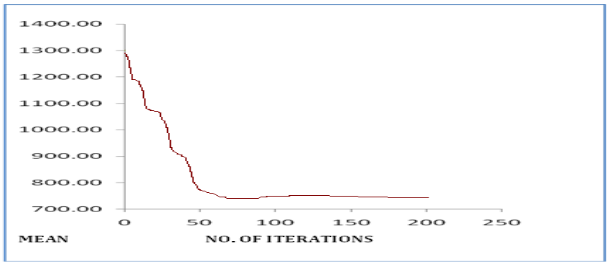

Figure.6. Convergence characteristics of FPSO+KHM

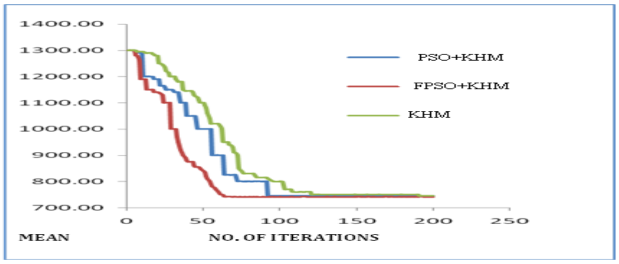

Figure.7. Convergence characteristics of KHM, PSO+KHM and FPSO+KHM for IEEE 300 Bus system

Figure 6 shows the convergence characteristics of Fuzzy PSO based K-Harmonics Means and it is observed that the mean value converges smoothly to the optimum (minimum) value without any abrupt oscillations approximately with 60 iterations due initialization and adaption of inertia weight produce fast convergence.

Figure 7 shows the convergence characteristics of FPSO+KHM , PSO+KHM and KHM and it is observed that , by fuzzified PSO proposed algorithm the mean value converges smoothly to the optimum (minimum) value without any abrupt oscillations and convergences quickly when compared to PSO+KHM and KHM algorithms.

Thus ensuring convergence reliability of proposed FPSO+KHM algorithm. And further it is observed that the FPSO has less number of iterations than KHM and PSO+KHM based algorithms.

6 CONCLUSION

The proposed Efficient Intelligent uses the algorithm uses Fuzzified PSO and K-harmonic means to generate more accurate, robust and better clustering and also generate the best solution in few number of iterations, avoid trapping in local optima and get faster convergence, when compare to KHM and hybrid PSO+ K Harmonic Means algorithm. This algorithm is applied for two different set of IEEE bus electrical data systems. The results show superior performance in terms of less Harmonic mean, Standard deviation, Iterations and CPU time when compared to PSO+KHM and KHM. Thus the proposed algorithm Fuzzified Particle Swarm Optimization based K- Harmonics Means is the best optimal solution for evaluating quality Clustering.

7. REFERENCES

[1] Gianfranco Chicco, Irinel-Sorin Ilie. Support Vector Clustering of Electrical Load Pattern Data. IEEE Transactions on Power Systems, 2009, 24(3):1619- 1628.

[2] Dubes R C, Jain A K.Clustering methodology in exploratory data analysis. In advances in computers, M.C . Yovits Ed, Academic Press, Inc., New York,1980,113-125.

[3] Berkhin P. Survey of clustering data mining techniques. Accrue Software Research Paper,2002,1-56.

[4] Anderberg M R.Cluster. Analysis for Applications. Academic Press, Inc., New York, 1973.

[5] Jain A K, Murty M N, Flynn P.Data Clustering- A Review.ACM Computing Survey, 1999, 31(3), 264-323.

[6] Sami Ayram, Tommi Kainen. Introduction to partitioning-based clustering methods with a robust example.Software and Computational Engineering,2006.

[7] Cui X, Potok T E. Document Clustering using Particle Swarm Optimization .IEEE Swarm Intelligence Symposium ,California,2005.

[8] [8] Jiawei Han, Micheline Kamber, Jian Pei .Data mining Concepts and Techniques. Elsevier Inc, 2012.

[9] Selim S Z, Ismail M A. K-means type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell,1984, 81–87.

[10] Osama Abu Abbas. Comparison between Data Clustering Algorithms. International Arab journal of information Technology, 2008, 5(3),320-325.

[11] Mehdi Neshat, Shima , Farshchian Yazdi, Daneyal Yazdani, Mehdi Sargolzaei. A New Cooperative Algorithm Based on PSO and K-Means for Data Clustering. Journal of Computer Science,2012, 8(2), 188-194.

[12] Bin Zhang. Generalized K-Harmonic Means- Boosting in Unsupervised Learning. Hewlett-Packard Laboratories, 2000.

[13] Bin Zhang, Meichum Hsu, Umeshwar Dayal. K-Harmonic Means A Data Clustering Algorithm: Hewlett-Packard Research Laboratories, 1999.

[14] Andries P Engelbrecht. Computational Intelligence: Wiley Publications, Second Edition, 2007.

[15] Jothi Swaroopan N M, Somasundaram P. Fuzzified PSO Algorithm for DC-OPF of Interconnected Power System. Journal of Theoretical and Applied Information Technology,2010,17(1),1-10.

[16] Rui Xu, Jie Xu, Donald C. Wunsch. A Comparison Study of Validity Indices on Swarm- Intelligence-Based Clustering. IEEE transactions on systems, man, and cybernetics—part b: cybernetics, 2010,42(4),1243-1256.

[17] [17]Hesam Izakian, Ajith Abraham . Fuzzy Clustering Using Hybrid Fuzzy c-means and Fuzzy Particle Swarm Optimization. IEEE- World Congress on Nature & Biologically Inspired Computing,2009,1690-1694.

[18] Fengqin Yang A B, Tieli Sun A, Changhai Zhang. An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization. Elsevier -Expert Systems with Applications,2009.

[19] Anirban Mukhopadhyay, Lopamudra DeyMicroarray. Gene Expression Data Clustering using PSO based K-means Algorithm. International Conference on Advanced Computing, Communication and Networks, 2011.

[20] Qian Xiao-Dong, Li Shi-Wei. Date Clustering using Principal Component Analysis and Particle Swarm Optimization. Dept. of Traffic & Transp., Lanzhou Jiaotong UnivChina,2010.

[21] Kennedy J, Eberhart R C. Shi Y. Swarm Intelligence. Morgan Kaufmann, New York, 2001.

[22] Shi Y H, Eberhart R C. Parameter Selection in Particle Swarm Optimization. The 7th Annual Conference on Evolutionary Programming, San Diego,1998.

[23] Bhubaneswar, Mishra D, Satapathy S K .Particle Swarm Optimization Based Fuzzy frequent Pattern Mining from Gene Expression Data. IEEE Explore Digital Library , 2011.

[24] Van D M, Engelbrecht A P.Data clustering using particle swarm optimization. Proceedings of IEEE Congress on Evolutionary Computation, Australia,2003, 215-220.

[25] Donghui Chen, Zhijing Liu, Zonghu Wang. A novel fuzzy clustering algorithm based on Kernel method and ParticlenSwarm Optimization. Journal of Convergence Information Technology,2012, 7(3), 299-307.

[26] Eberhart R C, Shi Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Congress on Evolutionary Computing,2000, 1(3), 84-88.

[27] Weimin Ma, Miaomiao Wang, Xiaoxi Zhu. Improved particle swarm optimization based approach for bilevel programming problem-an application on supply chain model. International Journal of Machine Learning and Cybernetics , Springer, 2013,4(2), 189-194.

8. AUTHOR’S PROFILE

A.Jaya Mabel Rani, Assistant Professor/CSE, Maamallan Institute of Technology, Chennai, India, was born in Kappikadu, KanyaKumari district, Tamilnadu, India. She did her MCA in Madras university and his ME (CSE) in Anna University, Chennai, Tamil Nadu. Presently she is pursuing her Ph.D research in Sathyabama University, Chennai. Her major fields are Data Mining, Data structure and Artificial Intelligence.