Ant Colony Optimization of Interval Type-2 Fuzzy C-Means with Subtractive Clustering and Multi-Round Sampling for Large Data

Sana Qaiyum[1], Izzatdin Aziz[2], Jafreezal Jaafar[3], Adam Kai Leung Wong[4]
Center for Research in Data Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia[1, 2, 3]
High Performance Computing Center, George Washington University, Washington, USA[4]

Abstract*—Fuzzy C-Means (FCM) is widely accepted as a clustering technique. However, it cannot often manage different uncertainties associated with data. Interval Type-2 Fuzzy C-Means (IT2FCM) is an improvement over FCM since it can model and minimize the effect of uncertainty efficiently. However, IT2FCM for large data often gets trapped in local optima and fails to find optimal cluster centers. To overcome this challenge an Ant Colony-based Optimization (ACO) is proposed. Another challenge encountered is determining the number of clusters to perform clustering. Subtractive clustering (SC) is an efficient technique to estimate appropriate number of clusters. Though for large datasets the convergence rate of ACO and SC becomes high and thus, it becomes challenging to cluster data and evaluate correct number of clusters. To encounter the challenges of large dataset, Multi-Round Sampling (MRS) technique is proposed. IT2FCM-ACO with SC and MRS technique performs clustering on subsets of data and determines suitable cluster centers and cluster number. The obtained clusters are then extended to the entire dataset. This eliminates the need for IT2FCM to work on the complete dataset. Thus, the objective of this paper is to optimize IT2FCM using ACO algorithm and to estimate the optimal number of clusters using SC while employing MRS to handle the challenges of voluminous data. Results obtained from several clustering evaluation measures shows the improved performance of IT2FCM-ACO-MRS compared to ITFCM-ACO and IT2FCM. Speed up for different sample size of dataset is computed and is found that IT2FCM-ACO-MRS is ≈1–5 times faster than IT2FCM and IT2FCM-ACO for medium datasets whereas for large datasets it is reported to be ≈ 30–150 times faster.

Keywords—Interval type-2 fuzzy c-means; ant colony optimization; subtractive clustering; multi-round sampling

I. Introduction

Clustering is the process of assigning a homogenous group of objects into subsets called clusters so that objects in each cluster are more similar to each other than objects from different clusters based on the values of their attributes [1]. Clustering technique has been studied extensively in various research areas like data mining [2, 3], pattern recognition [4], machine learning [5], image segmentation [6], semantic clustering [7] and membership function generation [8], [9]. Clustering is mainly divided into two main groups: hierarchical and partitioning algorithms. Partitioning clustering algorithms have been widely applied because of its efficiency and applicability for large data sets. The fuzzy clustering algorithm is currently widespread partitioning clustering algorithm. The FCM [10, 11] is commonly used technique for fuzzy clustering analysis because of its capability to handle uncertainty. FCM assign data object partially to multiple clusters with certain degree of membership and handle overlapping partitions. The degree of membership in fuzzy clusters depends on the closeness of the data object to the cluster centres. Although FCM is good in data clustering and has been the base for developing other clustering algorithms but is very susceptible to noise and incapable of handling large number of uncertainties associated with data set.

To tackle the issue of FCM algorithm efficiently, Hwang and Rhee proposed the combined use of Interval Type-2 Fuzzy logic technique [12] and FCM algorithm resulting in Interval Type-2 Fuzzy C-Means [13]. IT2FCM is an improvement over FCM that can model and minimize the effect of uncertainty more efficiently. The working principle of IT2FCM is like FCM. IT2FCM minimizes an objective function using an Alternating Optimization (AO) technique. IT2FCM randomly initializes either the membership matrix or cluster centres. Due to random initialization IT2FCM often gets trapped into local optimal solution and fails to return optimal values of cluster centroids [14–17]. It is also computationally expensive in terms of time and space for generating clusters for large datasets [18, 19].

The probability of finding global optima can be increased using bio-inspired metaheuristic techniques such as population, swarm-based or nature inspired algorithms. Several optimization techniques have been proposed to solve the problem at hand, but the focus of research has been FCM clustering algorithm while limited study has been found for optimizing IT2FCM. In this paper, an Ant Colony-based Optimization (ACO) technique has been proposed to optimize IT2FCM. ACO algorithm is a swarm-intelligence based bio-inspired technique that has been widely and successfully used for combinatorial optimization problems. ACO is based on the foraging behaviour of ants. ACO mimics the ability of indirect communication of ants to find the shortest path to food source by means of chemical pheromone trails. This characteristic of ants is exploited in ACO to solve various discrete optimization problem [20, 21]. Because of this inherent property of ACO, it has been used efficiently in FCM clustering to solve the problem of global optima [22, 23]. However, ACO optimization algorithm has not been introduced to solve the problem of IT2FCM algorithm. The use of ACO technique in IT2FCM is considered, owing to its ability for fast discovery of good solutions in discrete optimization problems and due to its adaptive nature in dynamic environment.

IT2FCM requires to input pre-estimated number of clusters “c” to perform clustering on the given dataset. To obtain the desirable cluster partitions in a given data, commonly c is set manually which is very subjective and arbitrary process. Several approaches have been proposed to select appropriate value of c. A rule of thumb was proposed where $c \leq N^{1/2}$, N is the data size and c are determined based on expert’s knowledge [24]. Another method is to determine c using cluster validity index such as Davies-Bouldin, Xie-Beni, and Dunn indices [19]. Subtractive clustering is another prevalent method to determine cluster number [24]. SC proposed by Chiu is a fast, one pass algorithm that can estimate the number of clusters c in a given dataset [25]. The value of c evaluated using SC method can be used to initialize IT2FCM. This will eliminate the task of manually feeding the number of clusters to the IT2FCM algorithm. However, for large data the convergence rate of SC is high and therefore it becomes very time consuming to determine number of clusters.

In clustering to handle the problem of large data two main approaches have been proposed; distributed clustering and clustering a sample determined by either progressive or random sampling [26]. Both methods offer useful techniques to achieve two main objectives: acceleration for loadable data and approximation for unloadable data. In this paper, to solve the problem of large data and further improve the performance of IT2FCM-ACO-SC, MRS technique has been proposed. MRS [27] is a straightforward approach, where samples of fixed size are generated using random sampling technique without replacement. Using MRS, IT2FCM does not need to perform clustering on the entire dataset but rather it obtains suitable clusters from samples of data. The results obtained from the samples are extended to the entire dataset, providing efficiency in terms of time and space.

The objective of this paper is to optimize IT2FCM using ACO to find optimal cluster centroids to improve the quality of clustering. SC with IT2FCM-ACO is used to obtain optimal number of clusters c. Further, to perform clustering efficiently and effectively in timely manner for large data, MRS technique is proposed. The paper is organized as follows: Section II discusses the related work; Section III gives an overview on background study of IT2FCM and ACO; Section IV presents proposed methodology along with their algorithms; Section V discusses the results obtained by comparing the proposed algorithm IT2FCM-ACO-MRS with IT2FCM-ACO and IT2FCM-AO using several evaluation metrics. Lastly, Section VI concludes this paper.

II. Related Work

In the literature, to optimize fuzzy clustering a variety of bio-inspired metaheuristic techniques have been proposed. These include population based: genetic algorithm (GA), teaching learning-based optimization (TLBO), differential evolution (DE); swarm-intelligence based: ant colony optimization, particle swarm optimization (PSO), artificial bee colony (ABC) and nature-inspired: simulated annealing (SA), and tabu search. Among these, ACO optimization algorithm has been successfully applied in clustering. A simplified version of ACO over original ant system algorithm was introduced that was used to solve the problem of Hard C-means(HCM) and Fuzzy C-means algorithm [23]. In another work, FCM-ACO algorithm was proposed for clustering suppliers into smaller groups with similar features [22]. All the proposed research works are focussed on optimizing FCM using ACO algorithm, however, no work has been found for IT2FCM. Further, all these studies do not take into consideration the volume of data.

In the context of IT2FCM limited study has been found regarding the optimization of IT2FCM to determine optimal initial cluster centroids. To overcome the problem of sensitivity to initial conditions Nguyen et al. [28] proposed a genetic IT2FCM (GIT2FCM) algorithm for the segmentation and classification of Multiplex Fluorescent In Situ Hybridization (M-FISH) images. It consists of two steps: firstly, the population of GA was randomly initialized and secondly, the cluster centroids were adjusted using GA based on cluster validity index determined by IT2FCM. For validation of the proposed method the results were compared with FCM, adaptive FCM (AFCM) and IT2FCM and the results prove that GA improves the performance of IT2FCM by determining appropriate cluster centroids.

IT2FCM is based on Euclidean norm which may not always be suitable for more general clusters. To overcome this issue Nguyen et al. [29] proposed an enhancement to IT2FCM by implementing multiple kernel-based method i.e. multiple kernel IT2FCM (MKIT2FCM). However, similar to IT2FCM, it had difficulty in determining the optimal values of cluster centres and number of clusters. To encounter these challenges the author [15] suggested GA based optimization to determine the optimal number of clusters and the initial cluster centroids. The result shows that GMKIT2FCM have high clustering quality than other algorithms such as KIT2FCM and MKIT2FCM. Though, GA is robust and powerful optimization algorithm for solving problems in complex search space [28] but often due to random initialization it suffers from premature convergence for large datasets [30]. Rubio and Castillo [31] implemented PSO optimization technique to IT2FCM, to automatically determine optimal number of clusters and interval-values of fuzzifier. For cluster evaluation, the simulation was conducted on synthetic dataset produced by Gaussian Mixture Method. The result shows that PSO enhances the performance of IT2FCM by identifying correct number of clusters and interval of fuzzification exponent. However, all these works do not cover large data environment.

To appropriately cluster large data several algorithms have been proposed in fuzzy clustering problem. Some of the widely used approaches are multi-round sampling [30], single-pass FCM (spFCM) [32], online FCM (oFCM) [33], bit-reduced FCM (brFCM) [34], kernel FCM (kFCM) [35], [36]. Among these algorithms, MRS is a fast approach for addressing large datasets. The kernel based FCM are suitable for estimating non-spherical clusters, however, it is computationally expensive. Similarly, brFCM is not suitable for multi-dimensional dataset. spFCM, oFCM and its other variants suffers from low performance compared to random sampling approach.

III. Background Study

A. Interval Type-2 Fuzzy C-Means

IT2FCM is an objective function-based clustering method used to minimize the distance between the input pattern and cluster prototype while determining the optimal value of cluster centroids and the membership matrix. A fuzzifier defines and manages the uncertainty to create an appropriate boundary of the fuzzy system. However, one fuzzifier cannot handle uncertainty for interval type-2 fuzzy sets; therefore two fuzzifier m_1 and m_2 were defined that represents different fuzzy degrees. Since, the fuzzifier value is represented by an interval $[m_1, m_2]$, the membership matrix $(\widetilde{U})$ and cluster centroids $(\widetilde{V})$ must be evaluated for the interval. IT2FCM minimizes an objective function $\tilde{J}_m$ as shown in (1).

$$J_m\left(\widetilde{U}, \widetilde{V}\right) = \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^{m} d_{ik}^{2} \tag{1}$$

where, m represents the two fuzzifier (m_1, m_2 >1), u_{ik} is the membership value of pattern x_i for cluster i, d_{ik}^2 is the distance between x_i and the cluster prototype v_k, c number of clusters between 2 and n-1, n total number of dataset, $\widetilde{U}$ represents membership matrix for the patterns x_k across each cluster with membership degree u_{ik} and $\widetilde{V}$ a matrix of a collection of all cluster prototypes v_k

For IT2FCM the region between the upper and lower memberships defines the footprint of uncertainty (FOU). Lower and upper membership matrices denoted by $\underline{u}_{ik}$ and $\overline{u}_{ik}$ given by (2) and (3) represents the lower and upper bound of FOU respectively. FOU implies the amount of uncertainty involved in the data. In IT2FCM the lower and upper membership matrix is randomly initialized in the interval [0,1] using Alternating Optimization (AO) method. Then it is used to update the lower and upper $\widetilde{V} = \left[\underline{v}_k, \overline{v}_k\right]$ cluster centroids as given by (4).

$$\overline{u}_{ik} = \max\left[\frac{1}{\sum_{j=1}^{c}\left(d_{ik}/d_{ij}\right)^{\overline{\frac{2}{(m_1-1)}}}}, \frac{1}{\sum_{j=1}^{c}\left(d_{ik}/d_{ij}\right)^{\overline{\frac{2}{(m_2-1)}}}}\right] \tag{2}$$

$$\underline{u}_{ik} = \min\left[\frac{1}{\sum_{j=1}^{c}\left(d_{ik}/d_{ij}\right)^{\overline{\frac{2}{(m_1-1)}}}}, \frac{1}{\sum_{j=1}^{c}\left(d_{ik}/d_{ij}\right)^{\overline{\frac{2}{(m_2-1)}}}}\right] \tag{3}$$

$$\widetilde{V} = \left[\underline{v}_k, \overline{v}_k\right] = \left[\frac{\sum_{i=1}^{n}(u_{ik})^{m_1} x_i}{\sum_{k=1}^{n}(u_{ik})^{m_1}}, \frac{\sum_{i=1}^{n}(u_{ik})^{m_2} x_i}{\sum_{i=1}^{n}(u_{ik})^{m_2}}\right] \tag{4}$$

where, $d_{ik}^2 = \|x_i - v_k\|$ is the distance between input patterns x_i and cluster centers v_k ($\|\cdot\|$is the Euclidean norm).

The values obtained for $(\widetilde{U}, \widetilde{V})$ are for the interval $[m_1, m_2]$ and therefore, must be type-reduced using (5) and (6) to obtain crisp values. This process continues until the cluster centres are stable or maximum iteration is reached.

$$v_k = \frac{\underline{v}_k + \overline{v}_k}{2} \tag{5}$$

and

$$u_{ik} = \frac{\underline{u}_{ik} + \overline{u}_{ik}}{2} \tag{6}$$

The structure of IT2FCM defined in this paper is based on the work of Rubio and Castillo [37].

B. Ant Colony Optimization

The fundamental concept of ACO [38] is based on the behaviour of ants in pursuit of food. In the real world, despite having limited vision, the ants can find the shortest path between their colony and the food sources by leaving down the pheromone trails along the shortest path. The pheromone trail starts to evaporate over time, this being an advantage if the path is no longer preferred.

The ACO algorithm duplicates this behaviour of ants by choosing solutions based on pheromones and updating pheromones based on the solution quality. Pheromone evaporation has the advantage to avoid local optima convergence. In this paper, the ACO algorithm proposed by Runkler [23] has been referred. The algorithm is described in Fig. 1.

```
input parameters $t_{max}, \rho, n$
$t_{max} \in N$ maximum number of iterations
$\rho \in [0,1]$ pheromone evaporation rate
$n$ size of dataset
Initialize pheromones $p_k = 1$, k=1, …., n
for $t = 1, \ldots, t_{max}$
    repeat
      randomly set $u_k = 1$ and $u_j = 0, j = 1, \ldots, n, j \neq k$,
      with probability $prob(k) = p_k / \sum_{j=1}^{c} p_j$
    until solution vector u is feasible
    compute objective function $J(u)$
    for j=1, …., n
        update pheromone $p_j = p_j . (1 - \rho) + u_j . f(J(u))$
    end for
end for
output solution u
```

Fig. 1. ACO Algorithm.

IV. Proposed Methodology

This section is divided into two subsections to clarify the proposed methodology. The first section describes ACO with SC to improve search for global optima and estimate cluster number in IT2FCM. Next section describes handling of large data of IT2FCM-ACO-SC algorithm using MRS technique.

A. IT2FCM-ACO

In IT2FCM AO algorithm is used to initialize membership matrix $\widetilde{U}$ and update cluster centroids $\widetilde{V}$ for each iteration while minimizing the objective function $\tilde{J}_m$. For the proposed methodology ACO algorithm based on Fig. 1 is introduced to minimize the objective function. Fig. 2, presents the proposed algorithm where the two fuzzifier m_1 and m_2 are considered whose value >1. In the proposed algorithm, each data pattern represents an ant in the real world and are allocated to one of the c clusters. The value of c clusters is predicted using SC algorithm. The allocation of data patterns is based on a pheromone matrix p. The basic idea is to randomly produce lower and upper membership matrix $\widetilde{U}$, whose expected values correspond to the normalized lower and upper pheromone matrix $\tilde{p} = [\underline{p}, \overline{p}]$ respectively. This is done by adding Gaussian noise with variance σ to the normalized matrix $\tilde{p}$. To keep $\widetilde{U}$ ∈ [0,1], the memberships are clipped at the borders of the interval [0,1], then normalized and finally checked for empty clusters. After initializing membership matrix, lower and upper values of cluster centroids ($\widetilde{V}$) and membership matrix ($\widetilde{U}$) are updated and type-reduced to get crisp values. Then objective function $\tilde{J}_m(\widetilde{U}, \widetilde{V})$ is minimized and the minimum value of objective function is computed. Then pheromone matrix $\tilde{p}$ is updated using values of $\widetilde{U}, \widetilde{V}$ *and* $\tilde{J}_m$ in each iteration. The algorithm continues until stable value of objective function is obtained or maximum iteration is reached.

B. IT2FCM-ACO with Multi-Round Sampling

The large dataset X is randomly divided into small samples S= {S_1, S_2, …., S_n} of fixed size. IT2FCM-ACO-SC rather than generating clusters for the complete data, performs clustering on samples of data. The samples are generated without replacement. IT2FCM-ACO is applied on the first sample S_1 to obtain values of membership matrix (U_{S_1}) and cluster centre (V_{S_1}), along with the value of number of clusters c_{S_1} using SC. Then, in next iteration IT2FCM-ACO-SC is applied on the next sample S_2. However, for the next iteration sample S2 is combined with S1 for clustering. IT2FCM-ACO produces new values of U_{S_2} and V_{S_2}; however, the values of centroids are initialized with the values of cluster centroids obtained from previous iteration. Moreover, for each iteration of new sample, the cluster number c is determined. The algorithm will terminate when the following conditions are satisfied: 1) when cluster centres obtained from previous and last iteration is less than the value of user-defined threshold (Ɛ) 2) the cluster number c does not vary from previous iteration. The values of membership matrix (U_s) and centroids (V_s) obtained from the sample sets are then extended for the entire dataset (X). Fig. 3 shows the flowchart of the proposed algorithm.

```
Initialize X, c, m1, m2
Compute cluster c using subtractive clustering method
where X= {$x_1$, $x_2$, ..., $x_n$}- data set, c- cluster number, $m_1$– 1.7 , $m_2$ – 2.6
Initialize $J_{min} = inf$
Initialize ACO parameters
$t_{max} - maximum\ iteration, t_{max} \in N$
$\rho \in [0,1]$ evaporation rate of pheromones
$\varepsilon > 0$ parameter is considered to avoid division by 0
$\alpha \geq 1$ varies the speed of convergence
min_impro – to check the variation in objective function from previous iteration
The values of the parameters are set based on literature review [23]
$t_{max}$ –1000, $\rho$ – 0.005, $\varepsilon$ –0.01, $\alpha$ –1.0, min_impro= 1e-5
Initialize pheromone matrix, $p_{ik} = 1,\ i = 1, \ldots., c\ and\ k = 1, \ldots, n$
for t=1 to $t_{max}$ do
    repeat
      for i=1 to n
        for k=1, …., c
          randomly set $\underline{u}_{ik} = \underline{p}_{ik}/\sum_{j=1}^{c} \underline{p}_{jk} + N(0,\sigma)$
          if $\underline{u}_{ik} < 0\ then\ \underline{u}_{ik} = 0$ end if
          if $\underline{u}_{ik} > 1\ then\ \underline{u}_{ik} = 1$ end if
          randomly set $\overline{u}_{ik} = \overline{p}_{ik}/\sum_{j=1}^{c} \overline{p}_{jk} + N(0,\sigma)$
          if $\overline{u}_{ik} < 0\ then\ \overline{u}_{ik} = 0$ end if
          if $\overline{u}_{ik} > 1\ then\ \overline{u}_{ik} = 1$ end if
        end for
        for k=1, …., c
          $\underline{u}_{ik} = \underline{u}_{ik}/\sum_{j=1}^{c} \underline{u}_{jk}$
          $\overline{u}_{ik} = \overline{u}_{ik}/\sum_{j=1}^{c} \overline{u}_{jk}$
        end for
      end for
    until $\sum_{i=1}^{n} \underline{u}_{ik} > 0\ and\ \sum_{i=1}^{n} \overline{u}_{ik} > 0,\ \forall 1 \leq i \leq c$
    compute centroids for lower ($\underline{v}_k$) and upper ($\overline{v}_k$) limit of the interval fuzzy clusters using (4)
    type reducing the interval of centroids using $v_k = \frac{\underline{v}_k + \overline{v}_k}{2}$
    compute lower ($\underline{u}_{ik}$) and upper ($\overline{u}_{ik}$) membership functions using (2) and (3)
    type reducing the interval of fuzzy partition matrix using $u_{ik} = \frac{\underline{u}_{ik} + \overline{u}_{ik}}{2}$
    calculate objective function $\tilde{J}_m(U,V) = \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^{m} d_{ik}^{2}$
    if $J < J_{min}^{m}$ then
        $J_{min}^{m} = J$
    end if
    for k=1, …., c
      for i=1, …., n
        Update lower(upper) pheromone matrix $\tilde{p}$
        $\underline{p}_{ik} = \underline{p}_{ik} \times (1 - \rho) + \underline{u}_{ik}/(J - J_{min} + \varepsilon)^{\alpha}$
        $\overline{p}_{ik} = \overline{p}_{ik} \times (1 - \rho) + \overline{u}_{ik}/(J - J_{min} + \varepsilon)^{\alpha}$
      end for
    end for
    if t > 1,
       if $|J(t) - J(t-1)| < \min\_impro$, break; end if
    end if
end for
```

Fig. 2. Proposed Algorithm of IT2FCM-ACO.

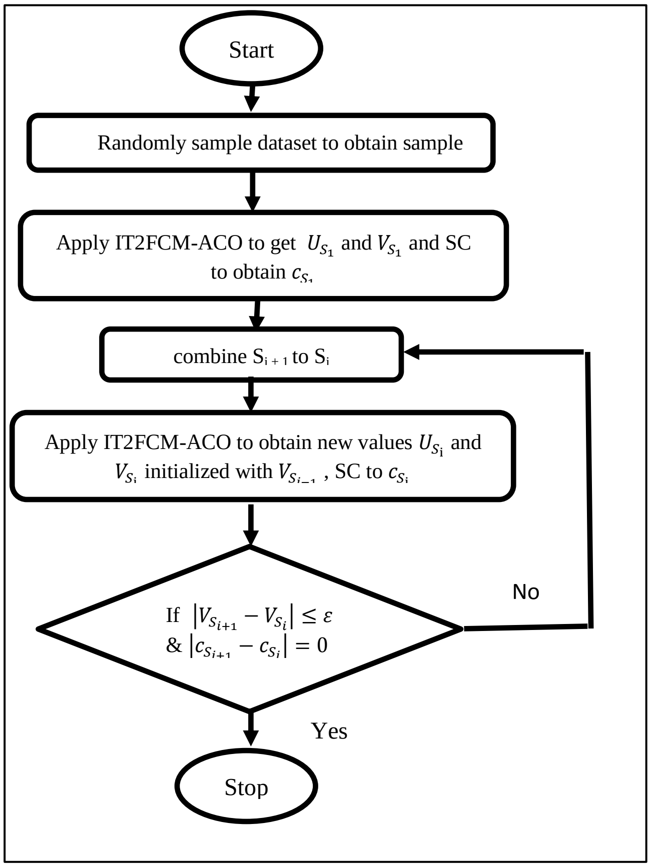
Start
Randomly sample dataset to obtain sample
Apply IT2FCM-ACO to get U_{S_1} and V_{S_1} and SC to obtain c_{S_1}
combine S_{i+1} to S_i
Apply IT2FCM-ACO to obtain new values U_{S_i} and V_{S_i} initialized with $V_{S_{i-1}}$, SC to c_{S_i}
If $|V_{S_{i+1}} - V_{S_i}| \leq \varepsilon$ & $|c_{S_{i+1}} - c_{S_i}| = 0$
No
Yes
Stop

Fig. 3. Flowchart of Proposed IT2FCM-ACO-MRS Technique.

V. Results and Discussion

In this section, the computational complexity of the proposed algorithm is computed and compared with IT2FCM-AO. The results obtained from different cluster validity index measures for IT2FCM-ACO and IT2FCM-ACO-MRS are discussed and compared with IT2FCM-AO. Also, the empirical analysis of algorithm efficiency in terms of speed up and memory is evaluated. The results reported in this paper are averages of 10 simulation runs. The algorithms are implemented in MATLAB R2017a on an Intel® Core™ i7 CPU @ 3.40 GHz with 8GB RAM.

A. Data Description

Huber, had proposed a classification of data by size as tiny, small, medium, large, huge and monster [39]. Later one more column was added and was categorized as very large [40]. The classified data set size is described in Table I. This has been set as standard to categorize the dataset used in this experiment. Table II gives an overview of the dataset used for the experiments.

B. Computational Complexity Analysis of Algorithm

The performance of an algorithm is evaluated in terms of computational complexity, which is the amount of resources necessary to execute an algorithm. The complexity of an algorithm is often computed in terms of time and space. Both complexities are denoted in terms of big-O.

1) Time complexity: To calculate time complexity only the highest order term of the expression is considered while ignoring any lower order terms. This is because the highest order terms have significant impact for large inputs. To determine the time complexity of the proposed method, the algorithm IT2FCM-ACO presented in Fig. 2 is divided into several steps. In step 1) three nested loops were run to initialize the value of lower and upper membership matrix. The first loop runs for number of dataset n and the next two loops run for cluster number c, time complexity can be approximated as $O(c^2.n)$. In step 2) loop is repeated until sum of rows of membership matrix is greater than 1 i.e. loop runs for c clusters. Since step 1 runs inside the loop described at step 2 the order of complexity becomes $O(c^3.n)$. In step 3) cluster centroids are computed for each clusters c using n data patterns for d dimension the time complexity becomes O(c.n.d). In step 3) membership matrix is updated by computing Euclidean distance for n rows, c columns and d dimension, therefore, order of complexity is O(c.n.d). In step 4) objective function is computed for n rows and c columns, thereby complexity is computed as O(c.n). In step 5) pheromone matrix is calculated thus, time complexity is estimated as O(c.n). Now the total computations (Adding all the 5 steps $O(c^3.n)$ + O(c.n.d) + O(c.n.d) + O(c.n) + O(c.n)) for single iteration is $O(c^3.n + c.n.d + c.n)$. Lower order terms are ignored. For maximum iteration t the time complexity is estimated as O(c3.n + c.n.d).t. If n>>d the order of complexity is further reduced to $\mathrm{O}(\mathrm{c}^3.\mathrm{n}+\mathrm{c.n}).\mathrm{t} \cong \mathrm{O}(\mathrm{c}^3.\mathrm{n.t})$.

TABLE I. Huber's Classification of Data Size

Bytes	**10^2**	**10^4**	**10^6**	**10^8**	**10^{10}**	**10^{12}**	**$10^{>12}$**
Size description	tiny	small	medium	large	huge	monster	Very large

TABLE II. An Overview of Dataset

Dataset	**Size**	**# attributes**	**# examples**	**# classes**
Weather[41]	medium	8	18,160	2
Electricity [42]	medium	8	45,312	2
Sea [41]	medium	3	10,00,001	2
Poker [43]	Large	10	10,25,010	10
Forest [44]	Large	54	5,81,012	7
Airlines [42]	Large	7	5,39,384	2

The time complexity of IT2FCM-AO is approximately computed as $O(c^2.n.t)$. Hence, the convergence rate of IT2FCM-ACO is higher to that of IT2FCM-AO. However, the higher time complexities of the two methods not necessarily results in higher run times. Therefore, the empirical analysis of run time and speed up is necessary and are presented in later section. For IT2FCM-ACO-MRS the dataset is divided into s number of samples, since the algorithm cluster a reduced set of data, the big-O time complexity has been reduced by s times. Time complexity for IT2FCM-ACO-MRS will be equivalent to $O((c^3.n.t)/s)$.

2) Space complexity: The complexity is determined by ignoring the space used by the inputs to the algorithm. Similar to time complexity, only the highest order terms are considered while the rest are ignored. For iterative loops, the variables or data structures that are declared apart from input will contribute to space complexity. To compute the complexity, first variables that are declared in the algorithm are identified. Seven matrices are found that were used in the algorithm for computation; lower and upper membership matrices of size (c,n), lower and upper cluster centroids of size (c,d), objective function (c,n) and pheromone matrices of dimension (c,n). Based on this, space complexity is computed as follows $O(c.n) + O(c.n) + O(c.d) + O(c.d) + O(c.n) + O(c.n) + O(c.n) + O(c.n)$. It will be reduced to the following form $5 * O(c.n) + 2 * O(c.d)$. Ignoring the constants space complexity will approximate to $O(c.n + c.d)$. The space complexity of IT2FCM-ACO is approximately equivalent to IT2FCM-AO. For IT2FCM-ACO-MRS samples of fixed size are extracted for each iteration. However, the samples are input to the program and thus it will not contribute to the space complexity. Since, IT2FCM-ACO-MRS converges for s samples of dataset, therefore results in reduced space complexity. The space complexity is computed as $\cong O((c.n + c.d)/s)$.

C. Simulation Results and Analysis

The performance of algorithms is analysed through several cluster validity index measures. These are divided into external and internal measures. External measures used in this paper are Davies-Bouldin (DB) Index and Dunn Index (DI) while external measures used are Purity, Rand Index (RI) and Error Rate (ER).

1) Cluster validity index measures: Through the simulation, it was found that determining number of clusters by employing SC algorithm is very time consuming for large datasets. Therefore, the results reported in Tables III-VII are for 20% of total dataset for poker, airlines and forest datasets.

Table III represents the value of Davies Bouldin (DB) [45] index for all the datasets for different algorithms. DB index measures how appropriately the data has been partitioned into clusters. A good clustering procedure estimates the value of DB index as low as possible. The lower the value of DB index indicates the object pairs within the same cluster are as close as possible i.e. compact although the clusters are well separated. From the table, it can be reported that value of DB index of IT2FCM-ACO is lower compared to IT2FCM-AO. This indicates that the distance between clusters centroid is less which results in low value of inter-cluster distance. Thus, it can be concluded that AO algorithm is not able to find appropriate cluster centroids. This results in excessive cluster overlapping. IT2FCM-ACO-MRS shows better results in most of the cases compared to both -ACO and -AO algorithms. Thus, proving its superiority over both the algorithms. Therefore, employing random sampling plus ACO based optimization technique to IT2FCM results in generating optimal cluster centroids and reduces the risk of proximity of cluster centroids.

Table IV shows the results of DI [10] which is another popular cluster evaluation measure. Higher values of DI indicate better clustering in the sense that the clusters are well separated and relatively compact. From the table, it is found that IT2FCM-ACO achieves high value of DI compared to IT2FCM-AO, thus indicates better clustering performance. However, IT2FCM-ACO-MRS attains relatively high values in comparison to both IT2FCM-ACO and -AO. Therefore, it can be concluded that the proposed algorithm partitions the data more efficiently and appropriately into clusters. The results obtained from both DB and DI shows the significance of ACO optimization to IT2FCM with MRS. Since, both the indices depend on inter- and intra-cluster distances, which in turn depends on the distance of data points from centroid or distance between the centroids. Therefore, optimal values of centroids are important to evaluate DB and DI. Hence, it can be stated that ACO produces optimal values of cluster centroids based on the results obtained from the two indices.

Table V presents the comparison of purity values obtained for different algorithms. Purity [46] is a simple cluster evaluation measure, that evaluates how close the obtained cluster is to the desired pure cluster. Poor clustering has purity value close to 0 while perfect clustering has values close to 1. The results obtained for IT2FCM-ACO are significantly higher compared to IT2FCM-AO. On the other hand, IT2FCM-ACO-MRS also shows significant improvement over IT2FCM-ACO.

Table VI compare the results of RI [47] for different algorithms. It is a measure of accuracy i.e. how accurately the given data points are partitioned into appropriate clusters. The value of RI lies between 0 and 1. Closer the value of RI to 1, more accurately the data points are clustered. IT2FCM-ACO achieves higher values of RI than IT2FCM-AO. From the table, it is observed that for large datasets such as poker, forest and airlines IT2FCM-ACO displays higher accuracy than medium datasets when compared to IT2FCM-AO. The algorithm IT2FCM-ACO-MRS attains high results over the other two algorithms. This signifies better clustering performance of the IT2FCM-ACO-MRS over IT2FCM-ACO and IT2FCM-AO.

TABLE III. EVALUATION OF DB VALUES FOR DIFFERENT ALGORITHMS

Dataset	**IT2FCM-AO**	**IT2FCM-ACO**	**IT2FCM-ACO-MRS**
Weather	0.8698	0.8338	0.8222
Electricity	2.0197	1.4333	1.4232
Sea	2.5671	2.4361	2.4293
Poker	322.48	173.60	165.32
Forest	438.56	434.33	424.32
Airlines	0.6962	0.6612	0.6584

TABLE IV. EVALUATION OF DI VALUES FOR DIFFERENT ALGORITHMS

Dataset	**IT2FCM-AO**	**IT2FCM-ACO**	**IT2FCM-ACO-MRS**
Weather	1.8630	1.9637	2.2990
Electricity	0.5342	1.2210	1.2086
Sea	0.7775	0.8208	0.9864
Poker	0.4352	0.9865	1.4235
Forest	0.0026	0.0047	0.0056
Airlines	1.5738	1.7899	1.8119

TABLE V. EVALUATION OF PURITY FOR DIFFERENT ALGORITHMS

Dataset	**IT2FCM-AO**	**IT2FCM-ACO**	**IT2FCM-ACO-MRS**
Weather	0.6786	0.6821	0.6836
Electricity	0.4145	0.4206	0.4269
Sea	0.7746	0.7865	0.8277
Poker	0.4229	0.5230	0.5832
Forest	0.7865	0.8685	0.8620
Airlines	0.4459	0.5235	0.5330

TABLE VI. EVALUATION OF RI MEASURE FOR DIFFERENT ALGORITHMS

Dataset	**IT2FCM-AO**	**IT2FCM-ACO**	**IT2FCM-ACO-MRS**
Weather	0.8505	0.8539	0.8513
Electricity	0.9195	0.9286	0.9287
Sea	0.8073	0.8641	0.8645
Poker	0.9227	0.9437	0.9558
Forest	0.7906	0.8368	0.8447
Airlines	0.8883	0.8903	0.8998

TABLE VII. EVALUATION OF ER FOR DIFFERENT ALGORITHMS

Dataset	**IT2FCM-AO**	**IT2FCM-ACO**	**IT2FCM-ACO-MRS**
Weather	22.2673	17.625	12.829
Electricity	8.725	3.393	2.376
Sea	7.046	6.925	4.622
Poker	44.459	44.230	32.813
Forest	38.2526	27.320	22.685
Airlines	47.051	45.235	23.177

Table VII illustrates the ER for different algorithms. ER gives the number of data points incorrectly assigned to the clusters. High value of ER indicates low performance of the algorithm while low value of ER indicates high performance compared to other algorithms. From the table, it is found that the ER obtained for IT2FCM-ACO is smaller than IT2FCM-AO, however, compared to IT2FCM-ACO-MRS the ER is high. Thus, its performance compared to the other two algorithms is high.

2) Computational efficiency analysis of an algorithm: The two most common measures to evaluate the algorithm efficiency are speed and memory usage. Speedup measures the relative performance of two algorithms and is computed in terms of practical run time. It is determined as the total amount of time spent to execute the function including its child functions. Memory usage is the space or the working memory (RAM) used by the algorithm.

Table VIII presents the comparison of run time and speedup computed for different algorithms. This table discusses the result obtained for IT2FCM-AO, IT2FCM-ACO and IT2FCM-ACO-MRS without implementing SC algorithm which is used to estimate the required number of clusters. For reasonable and easy evaluation of different algorithms, the number of clusters is set to 10 for all the datasets. From the table, it can be concluded that run time of IT2FCM-ACO is high compared to IT2FCM-AO for most of the datasets. As the size of data is increasing the run time for IT2FCM-ACO is increasing substantially. However, for sea and airlines dataset IT2FCM-AO has longer run time than IT2FCM-ACO. During simulation, it was found that IT2FCM-ACO converged in few iterations (sea, number of iterations t=305; airlines, t=589) while IT2FCM-AO (sea, t=562; airlines, t=1000) took larger number of iterations to converge. Also, from Table IX it is found that IT2FCM-ACO utilizes maximum memory during algorithm run compared to IT2FCM-AO. Therefore, to reduce the time and space complexity MRS technique is introduced. Since the time and space complexity depend on input size and MRS performs clustering on samples obtained from the entire dataset, therefore, it reduces the computational burden as well improve the cluster quality. This is evident from Tables VIII and IX where the run time and memory used by IT2FCM-ACO-MRS is significantly less compared to the other two algorithms.

The last two columns of Table VIII represent the speed up values of IT2FCM-ACO-MRS over IT2FCM-AO and IT2FCM-ACO respectively. Speed up $S_{-AO/-ACO-MRS}$ is the ratio of IT2FCM-AO and IT2FCM-ACO-MRS while $S_{-ACO/-ACO-MRS}$ is the ratio of IT2FCM-ACO and IT2FCM-ACO-MRS. For weather and electricity dataset the proposed method is at least 3 times faster than other two algorithms while for sea and poker dataset which is approximately of same dimension the speed of ≅1.6 is reported. For forest and airlines dataset (approximately equal number of data points) speed up between 4 and 5 is observed.

Table X evaluates the run time of algorithms for different percentage of dataset. These results are obtained by implementing SC algorithm to all the three algorithms. It is evident from the table that the run time of all the algorithms is increasing with the increase in sample size for all the datasets. For poker, forest and airlines dataset the run time of IT2FCM-AO-SC and -ACO-SC is increasing drastically as the size of the dataset is increasing. It is interesting to note that for medium size datasets (weather, electricity and sea) IT2FCM-ACO-SC takes longer time to execute compared to IT2FCM-AO-SC. However, for large datasets (poker, forest, and airlines) IT2FCM-ACO-SC takes less time to execute for each sample size compared to IT2FCM-AO-SC. Thus, IT2FCM-ACO-SC converges faster for large datasets in comparison to IT2FCM-AO-SC. Still, the run time for large datasets is considerably high for both the algorithms. For poker dataset, IT2FCM-AO-SC took about 1 hr to execute 20% of the complete dataset. For 100% sample size the run time is found to be ≈ 26 hours. However, during simulation the algorithm was not completely executed, only 60% of the algorithm was completed after 16 hours of continuous run of the algorithm. Therefore, the program was stopped, and the remaining run time was estimated. Similar results were obtained for forest dataset where the completion time is estimated to be 36 hours. It was found that the significant reason behind the longer run time for all the algorithms was high convergence rate of SC algorithm for large datasets. This is proven from Table IX where algorithms run time without SC for all the datasets is within an hour.

TABLE VIII. EVALUATION OF RUN TIME AND SPEED UP FOR DIFFERENT ALGORITHMS WITHOUT SC

Dataset	**Execution Time (sec)**			**Speedup**	
	IT2FCM-AO´	***IT2FCM-ACO***	***IT2FCM-ACO-MRS***	$S_{\text{-AO/ --ACO-MRS'}}$	$S_{\text{-ACO/-ACO-MRS}}$
Weather	9.951	19.356	3.277	3.036	5.906
Electricity	49.528	58.037	15.503	3.194	3.743
Sea	99.762	97.965	59.997	1.662	1.632
Poker	127.249	149.471	76.708	1.658	1.948
Forest	456.886	479.155	82.685	5.525	5.794
Airlines	1102.2	1079.23	267.239	4.124	4.038

TABLE IX. EVALUATION OF MEMORY FOR DIFFERENT ALGORITHMS WITHOUT SC

Dataset	**Memory (MB)**		
	IT2FCM-AO	***IT2FCM-ACO***	***IT2FCM-ACO-MRS***
Weather	1232	1236	1221
Electricity	1358	1380	1266
Sea	1289	1294	1241
Poker	2175	2399	1374
Forest	2133	2254	1564
Airlines	1635	1737	1323

TABLE X. EVALUATION OF RUN TIME FOR DIFFERENT SAMPLE SIZE WITH SC

Dataset	**Sample Percentage**					
		20%	***40%***	***60%***	***80%***	***100%***
Weather						
IT2FCM-AO	Run Time (sec)	1.94	3.74	5.83	16.24	24.86
IT2FCM-ACO		2.61	4.75	8.59	19.08	26.98
IT2FCM-ACO-MRS		2.46	2.54	3.90	4.82	8.70
Electricity						
IT2FCM-AO	Run Time (sec) Run	19.79	44.56	85.05	146.74	184.09
IT2FCM-ACO		67.27	101.90	104.64	204.84	490.60
IT2FCM-ACO-MRS		35.37	42.44	70.61	82.41	104.62
Sea						
IT2FCM-AO	Run Time (sec)	56.69	146.89	284.73	421.43	590.59
IT2FCM-ACO		99.31	237.06	415.13	578.48	792.96
IT2FCM-ACO-MRS		43.47	66.88	96.00	112.25	298.93
Poker						
IT2FCM-AO	Run Time (sec)	3954.35	15730.75	45128.56	76832.24	95997.60
IT2FCM-ACO		3786.06	14730.00	42163.24	74832.54	93484.60
IT2FCM-ACO-MRS		284.11	385.38	482.70	676.45	822.45
Forest						
IT2FCM-AO	Run Time (sec)	3954.35	15730.75	32567.39	47682.38	124658.60
IT2FCM-ACO		5604.02	20806.38	31826.38	45867.38	100058.60
IT2FCM-ACO-MRS		125.02	230.85	435.53	572.17	623.85
Airlines						
IT2FCM-AO	Run Time (sec)	740.97	3329.33	7635.92	15275.60	21159.17
IT2FCM-ACO		716.56	3164.48	7351.32	13242.62	20806.38
IT2FCM-ACO-MRS		26.00	44.71	69.76	133.47	245.25

Though airlines contain approximately the same number of examples as forest, but IT2FCM-AO-SC and -ACO-SC were able to execute the entire program in about 5 hrs. Similar pattern is observed for sea and poker dataset. The possible reason could be the increase in the number of attributes. Forest and poker dataset have 10 and 54 attributes respectively while sea and airlines contain only 3 and 7 attributes respectively. The dimension of the dataset can increase in two directions: number of variables and number of examples, thus from the results, it is proven that the multi-dimension dataset has significant impact on the convergence rate of SC algorithms.

To overcome the issue of high convergence rate MRS technique was proposed. In the proposed technique SC evaluates the required number of clusters for samples of dataset for each iteration until the program terminates. Hence SC does not need to determine the number of clusters for the entire dataset. From Table X it is noted that the proposed technique shows significant improvement for all the datasets compared to other two algorithms. The results are noteworthy for large datasets, where IT2FCM-ACO-SC-MRS can execute in lesser time. The reason behind the substantial increase in the performance of the proposed algorithm is that it can generate appropriate clusters within reasonable time for samples of data that is extended to the entire dataset without the need to perform clustering on the complete dataset.

Fig. 4 to 9 presents the speed up vs. sample size graph for different algorithms. It is evident from the graphs that speed of all the algorithms are decreasing as the sample size is increasing. In Fig. 4, for weather dataset IT2FCM-AO and -ACO at 20% sample size is 12 and 10 times faster respectively compared to 100%. A sudden decrease in the speed is observed from 20% to 40% sample size. For both the algorithms the speed has reduced t half compared to 20% sample size. The speed is decreasing drastically as the sample size is increasing, thus the execution time is increasing sharply. However, for IT2FCM-ACO-MRS the speed is decreasing steadily. This indicates that there is not much increase in the run time from 20% - 100% sample size.

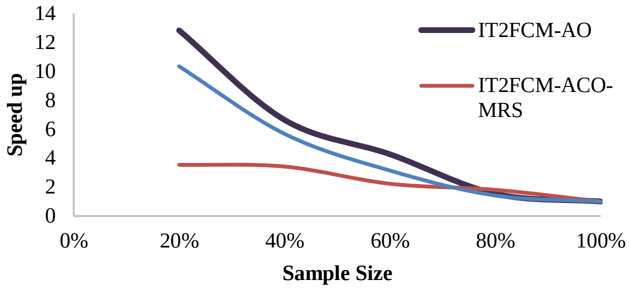

Fig. 4. Speed Up Vs Sample Size Evaluation of Weather Dataset for Different Algorithms.

In Fig. 5, a similar observation for electricity dataset is made. The speed is decreasing substantially for IT2FCM-AO and -ACO from 20% to 100% sample size while for IT2FCM-ACO-MRS the speed is reducing gradually. In Fig. 6, for sea dataset IT2FCM-AO, -ACO, -ACO-MRS at 20% sample size is 10, 7 and 6 times faster than 100% dataset respectively. For -AO and -ACO the decrease in speed is higher related to -ACO-MRS. For both IT2FCM-AO and -ACO the speed at 20% reduced from 10 and 7 to 4 and 3 at 40%, respectively. Though for IT2FCM-ACO-MRS the speed is decreasing at a slow pace compared to other two algorithms.

In Fig. 7, for poker dataset IT2FCM-AO and -ACO a sudden decrease from ≈ 24 to 6 is observed at 20% to 40% sample size. This suggests a high increase in run time. Although for IT2FCM-ACO-MRS the speed is almost linear suggesting with the increase in sample size the speed is decreasing consistently.

Similar observation is made for forest and airlines dataset in Fig. 8 and 9, respectively. Thus, it can be stated that for most of the datasets the speed of IT2FCM-ACO-MRS is consistent i.e. there is no drastic increase in the run time for the proposed algorithm.

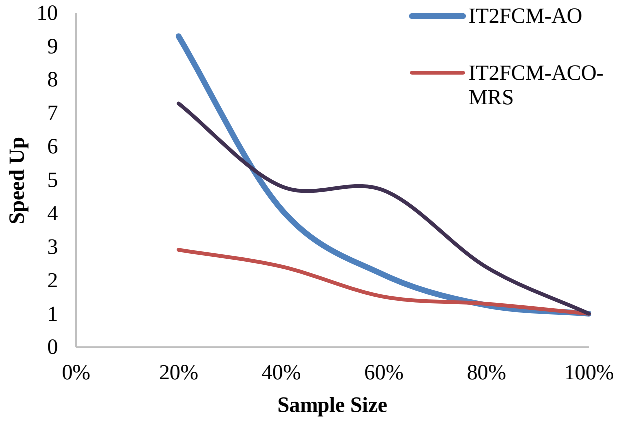

Fig. 5. Speed up Vs Sample Size Evaluation of Electricity Dataset for Different Algorithms.

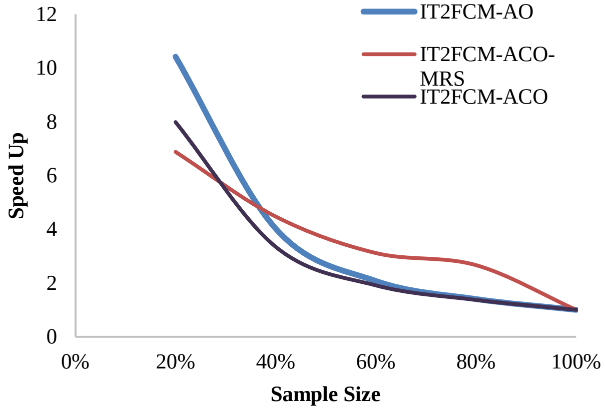

Fig. 6. Speed Up Vs Sample Size Evaluation of Sea Dataset for Different Algorithms.

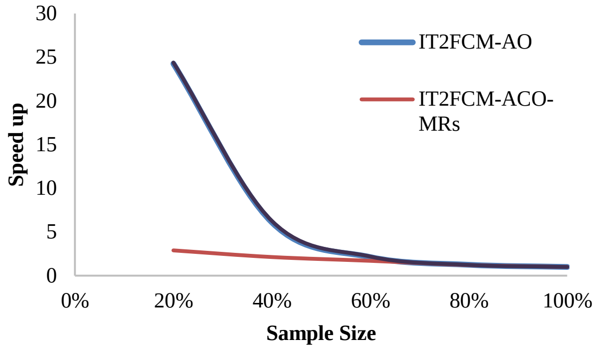

Fig. 7. Speed Up Vs Sample Size Evaluation of Poker Dataset for Different Algorithm.

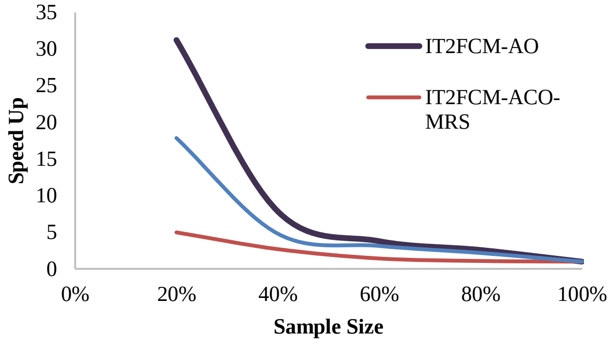

Fig. 8. Speed up Vs Sample Size Evaluation of Forest Dataset for Different Algorithms.

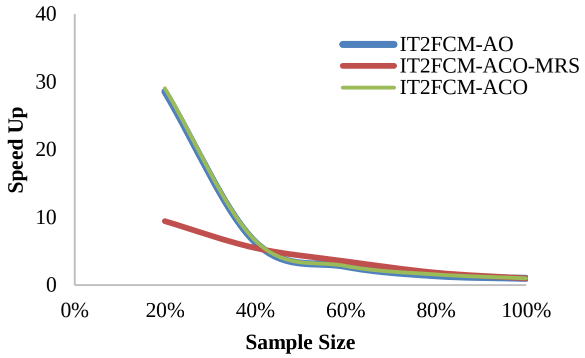

Fig. 9. Speed up Vs Sample Size Evaluation of Airlines Dataset for Different Algorithms.

Table XI mentions the speedup of IT2FCM-ACO-MRS over IT2FCM-AO and -ACO for different sample size. For weather, electricity and sea dataset IT2FCM-ACO-MRS is ≈ 1- 5 times faster than other two algorithms. The significant increase in the speed is observed for large datasets (poker, forest, and airlines). For poker and forest dataset it is seen that as the sample size is increasing the speed is also increasing, thus, IT2FCM-ACO-MRS is becoming faster compared to other two algorithms. For 20%, 40%, 60%, 80% and 100% IT2FCM-ACO-MRS is ≈ 13, 40, 90, 110, 116 times faster than -AO and -ACO respectively. The same result is found for forest and airlines dataset. These results prove the efficiency and competence of IT2FCM-ACO-MRS for clustering medium and large datasets.

TABLE XI. COMPARISON OF SPEED UP OF IT2FCM-ACO-MRS OVER IT2FCM-AO AND IT2-FCM-ACO

Dataset	**Sample Size**				
	20%	***40%***	***60%***	***80%***	***100%***
Weather					
$S_{IT2FCM-AO}$	5.223	2.613	1.092	3.371	2.859
$S_{IT2FCM-ACO}$	1.060	1.865	2.199	3.962	3.102
Electricity					
$S_{IT2FCM-AO}$	0.552	1.031	1.229	1.822	1.764
$S_{IT2FCM-ACO}$	1.877	2.357	1.512	2.544	4.701
Sea					
$S_{IT2FCM-AO}$	1.304	2.196	2.965	3.754	1.976
$S_{IT2FCM-ACO}$	2.284	3.544	4.324	5.153	2.652
Poker					
$S_{IT2FCM-AO}$	13.918	40.819	93.492	113.582	116.722
$S_{IT2FCM-ACO}$	13.325	38.222	87.349	110.626	112.329
Forest					
$S_{IT2FCM-AO}$	31.629	68.143	74.776	83.335	197.8991
$S_{IT2FCM-ACO}$	44.824	90.129	73.075	80.163	160.389
Airlines					
$S_{IT2FCM-AO}$	28.499	74.468	109.456	114.448	86.277
$S_{IT2FCM-ACO}$	27.561	70.780	105.3771	99.216	84.839

VI. CONCLUSIONS

This paper presents an improved IT2FCM clustering algorithm based on ACO optimization technique. This algorithm utilizes the global search property of ACO to estimate optimal cluster centres. Thus, overcomes the problem of IT2FCM returning locally optimum value. To eliminate the issue of manual feeding of cluster numbers SC is implemented to IT2FCM-ACO. SC extracts the expected number of clusters from the data itself and feed the information to ACO algorithm. However, ACO and SC algorithms have high convergence rate for large data. Thereby, to solve this issue MRS technique is proposed. It gives IT2FCM scalable approach as it eliminates the need for availability of entire dataset for clustering. Thus, it improves upon the time and space complexity of IT2FCM-ACO-SC.

With reference to DB and DI, it has been proven that IT2FCM-ACO-MRS with SC produces compact and well-separated clusters. The results obtained from purity, RI, and ER proves the high clustering performance of the proposed algorithm in comparison to IT2FCM-AO and -ACO. Further, the computational complexity in terms of time and space of the three algorithms are computed. From the result, it is found that IT2FCM-ACO-SC has high convergence rate for large datasets where it can take about hours to execute. However, when implemented with MRS technique it considerably reduces the time and space during algorithm run. The results obtained for run time and speed up proves the significant improvement of IT2FCM-ACO-SC-MRS over IT2FCM-ACO-SC and IT2FCM-AO-SC for both large and medium datasets. Further, the big-O computational analysis of the algorithms approves the advantage of combining ACO-SC with MRS to generate appropriate number of clusters for large data.

The proposed technique shows significant enhancement over traditional clustering technique for large datasets. However, for data stream environment where the voluminous data may be coming continuously and most likely boundlessly over time and may evolve over time. Such data stream environment may require incremental approach to capture the significance of new incoming data. The incremental technique processes data in chunks which improves upon time and space complexity. However, MRS method only works upon a sample of data and thus, may not be able to partition new incoming data into appropriate clusters. Therefore, for future work authors propose an incremental approach to IT2FCM-ACO to capture the characteristics of data stream environment.

ACKNOWLEDGMENT

This work was supported by Yayasan Universiti Teknologi PETRONAS (UTP) fund under Grant 0153B2-E67, at Centre for Research in Data Science (CeRDaS).

REFERENCES

[1] K. Jain, “Data clustering: 50 years beyond K-means,” Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, 2010.

[2] S. Qaiyum, I. A. Aziz, and N. Haron, “Quality-of-Experience modeling in high-density wireless network,” J. Adv. Res. Des., vol. 14, no. 1, pp. 10–27, 2015.

[3] S. Qaiyum, I. A. Aziz, and J. Bin Jaafar, “Analysis of big data and quality-of-experience in high-density wireless network,” in 3rd Int. Conf. on Computer and Information Sciences (ICCOINS ’16), 2016, pp. 287–292.

[4] B. I. Choi and F. Chung-Hoon Rhee, “Interval type-2 fuzzy membership function generation methods for pattern recognition,” Inf. Sci. (Ny)., vol. 179, no. 13, pp. 2102–2122, 2009.

[5] M. Yuwono, S. W. Su, B. D. Moulton, and H. T. Nguyen, “Data clustering using variants of rapid centroid estimation,” IEEE Trans. Evol. Comput., vol. 18, no. 3, pp. 366–377, 2014.

[6] C. Qiu, J. Xiao, L. Yu, L. Han, and M. N. Iqbal, “A modified interval type-2 fuzzy c-means algorithm with application in MR image segmentation,” Pattern Recognit. Lett., vol. 34, no. 12, pp. 1329–1338, 2013.

[7] W. Mohammad, A. Mohammad, I. Aziz, and J. Jaafar, “A survey on textual semantic classification algorithms,” in IEEE Conf. on Big Data and Analytics, 2017, pp. 1–6.

[8] F. C.-H. Rhee and B.-I. Choi, “Interval type-2 fuzzy membership function design and its application to radial basis function neural networks,” 2007 IEEE Int. Fuzzy Syst. Conf., vol. 1, no. c, pp. 1–6, 2007.

[9] E. Rubio, O. Castillo, F. Valdez, P. Melin, C. I. Gonzalez, and G. Martinez, “An extension of the fuzzy possibilistic clustering algorithm using type-2 fuzzy logic techniques,” Adv. Fuzzy Syst., vol. 2017, pp. 1–23, 2017.

[10] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” J. Cybern., vol. 3, no. 3, pp. 32–57, 1973.

[11] J. C. Bezdek, “Pattern recognition with fuzzy objective function algorithms,” SIAM Rev., vol. 25, no. 3, pp. 442–442, 1983.

[12] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning-I,” Inf. Sci. (Ny)., vol. 8, no. 3, pp. 199–249, 1975.

[13] C. Hwang and F. C.-H. Rhee, “Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means,” IEEE Trans. Fuzzy Syst., vol. 15, no. 1, pp. 107–120, 2007.

[14] D. D. Nguyen, L. T. Ngo, and L. T. Pham, “Genetic based interval type-2 fuzzy c-means clustering,” in Context-Aware Systems and Applications, 2013, pp. 239–248.

[15] D. D. Nguyen, L. T. Ngo, and L. T. Pham, “GMKIT2-FCM : A genetic-based improved multiple kernel interval type-2 fuzzy c-means clustering,” in 2013 IEEE International Conference on Cybernetics (CYBCO), 2013, pp. 104–109.

[16] L. Wang, Y. Zhang, and M. Cai, “The Global Interval Type-2 Fuzzy C-Means Clustering Algorithm,” in International Conference on Multimedia Technology, 2011, pp. 2694–2697.

[17] H. Liu, F. Zhao, and V. Chaudhary, “Pareto-based interval type-2 fuzzy c -means with multi-scale JND color histogram for image segmentation,” Digit. Signal Process., vol. 76, pp. 75–83, 2018.

[18] J. K. Parker and L. O. Hall, “Accelerating fuzzy-c means using an estimated subsample size,” IEEE Trans. Fuzzy Syst., vol. 22, no. 5, pp. 1229–1244, 2014.

[19] J. Zhang and L. Shen, “An improved fuzzy c-means clustering algorithm based on shadowed sets and PSO,” Hindawi Publ. Corp. Comput. Intell. Neurosci., vol. 2014, no. 60873247, p. 10, 2014.

[20] R. F. Tavares Neto and M. Godinho Filho, “Literature review regarding ant colony optimization applied to scheduling problems: guidelines for implementation and directions for future research,” Eng. Appl. Artif. Intell., vol. 26, no. 1, pp. 150–161, 2013.

[21] B. Chandra Mohan and R. Baskaran, “A survey: ant colony optimization based recent research and implementation on several engineering domain,” Expert Syst. Appl., vol. 39, no. 4, pp. 4618–4627, 2012.

[22] W. Liu and L. Jiang, “A clustering algorithm for supplier base management,” in Advanced Data Mining and Applications, 2010, pp. 106–113.

[23] T. A. Runkler, “Ant colony optimization of clustering models,” Int. J. Intell. Syst., vol. 20, no. 12, pp. 1233–1251, 2005.

[24] J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram, Fuzzy models and algorithms for pattern recognition and image processing. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[25] S. L. Chiu, “Fuzzy model identification based on cluster estimation,” J. Intell. Fuzzy Syst., vol. 2, no. 3, pp. 267–278, 1994.

[26] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, “Fuzzy c-means algorithms for very large data,” IEEE Trans. Fuzzy Syst., vol. 20, no. 6, pp. 1130–1146, 2012.

[27] M. Hung and D. Yang, “An efficient fuzzy c-means clustering algorithm,” in IEEE International Conference on Data Mining, 2001.

[28] D. D. Nguyen, L. T. Ngo, and J. Watada, “A genetic type-2 fuzzy c-means clustering approach to M-FISH segmentation,” J. Intell. Fuzzy Syst., vol. 27, pp. 3111–3122, 2014.

[29] D. D. Nguyen and L. T. Ngo, “Multiple kernel interval type-2 fuzzy c-means clustering,” in IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2013.

[30] Y. Xianfeng and L. Pengfei, “Tailoring fuzzy c-means clustering algorithm for big data Using random sampling and particle swarm optimization,” Int. J. Database Theory Appl., vol. 8, no. 3, pp. 191–202, 2015.

[31] E. Rubio and O. Castilo, “Optimization of the interval type-2 fuzzy c-means using particle swarm optimization,” in Nature and Biologically Inspired Computing (NaBIC), 2013, pp. 10–15.

[32] P. Hore, L. O. Hall, and D. B. Goldgof, “Single pass fuzzy c means,” 2007 IEEE Int. Fuzzy Syst. Conf., pp. 1–7, 2007.

[33] P. Hore, L. O. Hall, D. B. Goldgof, Y. Gu, A. A. Maudsley, and A. Darkazanli, “A scalable framework for segmenting magnetic resonance images,” J. Signal Process. Syst., vol. 54, no. 1–3, pp. 183–203, 2009.

[34] S. Eschrich, L. O. Hall, and D. B. Goldgof, “Fast accurate fuzzy clustering through data reduction,” IEEE Trans. Fuzzy Syst., vol. 11, no. 2, pp. 262–270, 2003.

[35] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, “Approximate kernel k-means: solution to large scale kernel clustering,” Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’11, p. 895, 2011.

[36] T. C. Havens, R. Chitta, A. K. Jain, and R. Jin, “Speedup of fuzzy and possibilistic kernel c-means for large-scale clustering,” in IEEE International Conference on Fuzzy Systems, 2011, pp. 463–470.

[37] E. Rubio and O. Castillo, “Interval type-2 fuzzy clustering for membership function generation,” in 2013 IEEE Workshop on Hybrid Intelligent Models and Applications (HIMA), 2013, pp. 13–18.

[38] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” IEEE Trans. Syst. Man, Cybern. Part B Cybern., vol. 26, no. 1, pp. 29–41, 1996.

[39] P. J. Huber, Data analysis: What can be learned from the past 50 years. 2011.

[40] R. J. Hathaway and J. C. Bezdek, “Extending fuzzy and probabilistic clustering to very large data sets,” Comput. Stat. Data Anal., vol. 51, no. 1, pp. 215–234, 2006.

[41] R. Elwell and R. Polikar, “No Title.” [Online]. Available: http://users.rowan.edu/~polikar/research/NSE/.

[42] “Massive online analysis.” [Online]. Available: https://moa.cms.waikato.ac.nz/datasets/.

[43] J. Alcala-Fdez et al., “KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework,” J. Mult. Log. Soft Comput., vol. 17, no. 2–3, pp. 255–287, 2011.

[44] D. Dheeru and K. T. Efi, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[45] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, no. 2, pp. 224–227, 1979.

[46] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to information retrieval. New York, NY, USA: Cambridge University Press, 2008.

[47] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” J. Am. Stat. Assoc., vol. 66, no. 336, pp. 846–850, 1971.