ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 3, 2014, *pp*.196-209

A Hybrid Data Clustering Approach Based on Cat Swarm Optimization and K- Harmonic Mean Algorithm

Yugal Kumar and G. Sahoo
Department of Information Technology, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India.
yugalkumar@bitmesra.ac.in
Department of Information Technology, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India.
gsahoo@bitmesra.ac.in

(*Received November 5, 2013, accepted May 21, 2014*)

Abstract. Clustering is an important task that is used to find subsets of similar objects from a set of objects such that the objects in the same subsets are more similar than other subsets. Large number of algorithms has been developed to solve the clustering problem. K-Harmonic Mean (KHM) is one of the popular technique that has been applied in clustering as a substitute of K-Means algorithm because it is insensitive to initialization issues due to built in boosting function. But, this method is also trapped in local optima. On the other hand, Cat Swarm Optimization (CSO) is the latest population based optimization method used for global optimization. In this paper a hybrid data clustering method is proposed based on CSO and KHM which includes the advantage of both algorithms and named as CSOKHM. The hybrid CSOKHM not only improved the convergence speed of CSO but also escape the KHM method to run in local optima. The performance of the CSOKHM is evaluated using seven datasets and compared with KHM, PSO, PSOKHM, ACA, ACAKHM, GSAKHM, CSO methods. The experimental results show the applicability of CSOKHM method..

Keywords: Cat swarm optimization, Data clustering, K-harmonic means, Gravitational search algorithm, Particle swarm optimization

1. Introduction

Clustering is an essential tool in pattern recognition, data mining and machine learning domain. It is NP Complete problem to find out hidden patterns, knowledge and information from a dataset that is previously unknown using some criterion function [1]. In clustering, a dataset is divided into K number of groups. The elements in one group are more similar to another group. K-Means (KM) algorithm is the oldest algorithm that has been widely used in clustering domain to find optimal cluster centers in datasets [2]. This algorithm is simple, fast and efficient but suffered with initialization and local optima problem [3]. Hence, to overcome the problems of KM algorithm and improve the efficiency of the clustering, hybridize version of KM algorithms have been developed by various researchers [4]. Instead the hybridization of KM, Zhang et al. [5] has developed K-Harmonic Means (KHM) algorithm for data clustering. In KHM, the clustering objective is to minimize average harmonic means to all instances of dataset in lieu of average mean (KM) from all cluster centers. The KHM algorithm has provided better result in comparison of KM but this algorithm is also suffered with stuck in local optima problem. In recent years, numbers of algorithms based on swarms, insects and natural phenomena's have been developed by researchers to solve clustering problem such as ABC [6], ACO [7], GA [8], PSO [9], CSO [10], BH [11], GSA [12] and many more. These algorithms are categorized as swarm based algorithms, biological based algorithms and basic science based algorithms. The above mentioned algorithms have immense potentials over prevailing traditional methods but these methods have suffered with several problems, for instance GA suffers from population diversity problem and the quality of solutions in GA depends on mutation and crossover probability [13]. The convergence time of ACO method is uncertain and probability distribution function change in each iteration [14]. PSO algorithm has weak exploitation property and sometimes stuck in local optima [15]. The performance of ABC algorithm is depended on the dimension of problem as dimension of problem is increased the convergence speed of ABC is decreased [16]. The GSA algorithm is sometimes suffered with premature convergence and there is no recovery if premature convergence exists because GSA is memory less algorithm [17].

The cat swarm optimization (CSO) is the latest, state of art animal inspired algorithm developed by Chu et al. [18], observing the behavior of cats. CSO is the first algorithm based on the behavior of animals as reported in the literature. The animal inspired algorithms are the sub branch of swarm based algorithms. CSO algorithm has been applied in many areas and provides remarkable results [19, 20, 21]. The main advantage of the CSO algorithm is good exploration property. Hence in this paper, a hybrid data clustering algorithm is proposed based on the CSO and KHM, to escapes the KHM run in local optima problem and increases the convergence speed of CSO. The performance of proposed algorithm is tested on several benchmark datasets which are downloaded from UCI repository and the proposed algorithm is more accurate and precise than others. The rest of the paper is organized as follow. Section 2 introduces KHM algorithm. Section 3 describes CSO technique. Section 4 presents hybrid CSOKHM clustering algorithm. Section 5 illustrates investigational results. Finally, section 6 gives conclusions.

2. K Harmonic Algorithm

The KHM is partition based iterative algorithm that evaluates the cluster using K centers. KMH is unconcerned to the initialization issues and provides faster convergence than KM when the initial cluster points far from local optimal. In case of KM, quality of solution depends on the initial cluster centers [22]. In KHM, the distance among instances of dataset to cluster centers are calculated by harmonic means. The performance of KHM algorithm is evaluated using the following equation.

$$KHM(X,C)=\sum_{i=1}^{N}\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|x_i-c_j\right\|^{p}}} \tag{1}$$

$\mathrm{X}=\{\mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3,\ldots\ldots,\mathrm{x}_\mathrm{n}\}$: Data instance for clustering.

$\mathrm{C}=\{\mathrm{c}_1,\mathrm{c}_2,\ldots\ldots,\mathrm{c}_\mathrm{k}\}$: Number of clusters.

$\mathrm{m}(\mathrm{c_j}/\mathrm{x_i})$: membership function to define the data point $\mathrm{x_i}$ belongs with center $\mathrm{c_j}$.

$\mathrm{w}(\mathrm{x_i})$: weight function to measure the influnce of data instances $\mathrm{x_i}$ to recompute the cluster centers.

The steps of KHM clustering algorithm can be given as

- Randomly initialize the cluster centers.
- Evaluate the value of objective function using equation 1.
- For each data instance x_i ,
    - Calculate its membership function *m(* $c_j/x_i)$ from all cluster centers using given equation

$$m(c_j/x_i)=\frac{\left\|x_i-c_j\right\|^{-p-2}}{\sum_{j=1}^{k}\left\|x_i-c_j\right\|^{-p-2}} \tag{2}$$

- Calculate the weight $w(x_i)$ using following equation

$$w(x_i)=\frac{\sum_{j=1}^{k}\left\|x_i-c_j\right\|^{-p-2}}{\left(\sum_{j=1}^{k}\left\|x_i-c_j\right\|^{-p}\right)^{2}} \tag{3}$$

- For each cluster center, recalculate the cluster centers from all data instances x_i with the help of memberships and weights respectively

$$c_j = \frac{\sum_{i=1}^{n} m(c_j/x_i)w(x_i)x_i}{\sum_{i=1}^{n} m(c_j/x_i)w(x_i)} \tag{4}$$

- Repeat the steps 2- 4, until the center points do not change considerably.
- Assigned the data instance x_i to cluster j with the biggest m (c_j/x_i).

3. Cat Swarm Optimization Algorithm (CSO)

ster center, recalculate t

Chu and Tasi has developed CSO method in 2007 by observing the behavior of cats [18]. The CSO method has developed on the unique property of cats i.e., acting while resting. Therefore on above mentioned property of cats, the behavior of cats is measured in two modes- in acting state and in resting state. The representation of solution set is the key property of optimization methods. Hence in CSO, the positions of the cats are used to evaluate the solution sets. Hence the CSO method consist of two modes- Seeking Mode and Tracing Mode. The detailed descriptions of these modes are given below:

- Seeking Mode: The seeking mode of the CSO method is defined as rest and alert. Its means a cat is always in alert position when it is resting. Hence in seeking mode, a cat is continually changed its position and try to achieve better position. The position of cats has changed according to the fitness function. The fitness function of CSO method is defined by equation 6. The seeking mode of CSO method is consisted of the following four parameters.
  i) Seeking Memory Pool (SMP)
  ii) Seeking Range of selected dimension (SRD)
  iii) Counts of dimension change (CDC)
  iv) Self position consideration (SPC)

$$Fit_i = \frac{\|FS_i - FS_k\|}{FS_{max} - FS_{min}} \qquad where\ 0 < i < j \tag{5}$$

- Tracing Mode: The tracing mode of CSO method is correspondent to the movement of cat i.e. how the cat traced the targets. So in tracing mode, the cats move according to its velocity in each dimension and update its position.
- The velocity of cats and its updated position is calculated by given equation:

$$v_{k,d} = v_{k,d} + r_1.c_1.\left(x_{best,d} - x_{k,d}\right) \qquad where \quad d = 1,2,3,\dots\dots\dots\dots\dots.., M \tag{6}$$

$$x_{k,d} = x_{k,d} + v_{k,d} \tag{7}$$

The following notations are used to devise the CSO method:

$x_{best,d}$: Best position of cat in d dimensional space
$x_{k,d}$: Position of Cat_k
$v_{k,d}$: Velocity of Cat_k

r_1 : Random value in the range of [0, 1]
c_1 : Constant

Mixture Ratio (MR) is used to combine the seeking and tracing mode of CSO method. The MR is also used to determine how many cats are used in seeking mode and tracing mode.

Steps of CSO method:

- Initialize the population of Cats.
- Define the parameters and specify the numbers of cat for seeking mode as well as tracing mode according to MR.
- Evaluate the value of fitness function for each cat to determine the position and memorize the best position.
- According to the flag:
  - If Cat_k is in seeking mode, apply the seeking mode process.
  - Otherwise, apply tracing mode process.
- Again set the number of cats to tracing and seeking mode according the value of MR.
- Repeat steps 3- 5 until the termination condition is satisfied and exit.

4. Proposed Hybrid CSOKHM Algorithm

The CSO method has good exploration property that's why it always meets the global optimal solution. But this method takes large computation time. The computational time of CSO is depanded on the value of SMP parameter; greater the value of SMP, larger the time required by the algorithm for execution [19]. On the other hand the KHM is fast, efficient and required less number of function evaluations but it suffers with local optima problem. Therefore, the proposed hybrid algorithm includes the advantage of both CSO and KHM, called it CSOKHM. The main objective of proposed algorithm is to escape the KHM from stuck in local optima and improve the convergence speed of CSO. However, the proposed CSOKHM uses the fitness function of KHM that is described in equation 1. In CSO method, cluster centers are represented by the cats and the positions of the cats provide the solution set. Before applying the CSO method in clustering problem, few adjustments have been made in original CSO method [10].

- The MR is removed such that every cat will be moved in seeking as well as tracing mode.
- The CDC parameter is also removed in tracing mode such that every dimension of cat's copy will be changed.

The description of the CSOKHM algorithm is given below and the corresponding flow chart of the CSOKHM algorithm is shown in Figure 1.

Step 1: Set the initial parameter population size and numbers of cluster centers, c_1, r_1, SMP, SRD, SPC, maximum iteration, iteration count.

Step 2: Initialize a population of size Cat_k

Step 3: Set iterative count Gen1 = 0.

Step 4: Set iterative count Gen2 = Gen3 = 0, calculate the value of objective function and fitness function

Step 5: Start the seeking mode of CSO method, for each cat_k, do following:

Step 5.1: Make the copy of present cluster center (i.e. cat_k) position to the value of SMP.

Step 5.2: Determine how many copy of present cluster center will be mutated.

Step 5.3: Compute the value of mutated copy using (SRD * present cluster center).

Step 5.4: Repeat the step 5.1, 5.2 and 5.3, until all cluster centers will be discovered.

Step 5.5: Randomly add and minus initial clusters from mutated cluster centers

Step 5.6: Calculate the objective function and fitness function for the newly generated cluster centers.

Step 5.7: Select new cluster center based on fitness function .

Step 6: Start the tracing mode of CSO method, for each cat_k, do following:

Step 6.1: Update velocity of cat_k using equation 6.

Step 6.2: Update the position of cat_k using equation 7 and evaluate the new cluster ceters.

```
        Step 6.3: Calculate the objective function and Fitness function for newly generated cluter
                  centers.
Step 7: Compare the fitness function of seeking mode and tracing mode
        Step 7.1: If {fitness function (Seeking mode) > fitness function (Tracing mode)}
        Step 7.2: Use cluster center of seeking mode.
        Step 7.3: Else, cluster center of tracing mode.
        Step 7.4: Gen2=Gen2+1. If Gen2<8, go to Step 5.1.
Step 8: For each $cat_k$ (KHM)
        Step 8.1: Initial cluster centers of the KHM algorithm are position of $cat_k$.
        Step 8.2: Recompute each cluster center using the KHM algorithm.
        Step 8.3: Gen3=Gen3+1. If Gen3<4, go to Step 8.1.
Step 9: Gen1= Gen1+1. If Gen1<Iteration Count, go to Step 4.
Step 10: Assign data instance $x_i$ to cluster k with the leading $m\ (c_j/x_i)$.
```

5. Experimental Results

To test the effectiveness of the proposed algorithm, it is applied on seven datasets which contain two artificial datasets i.e., synthetic dataset1 and dataset 2 and five benchmark datasets. The performance of proposed algorithm is compared with KHM, PSO, PSOKHM, GSAKHM, ACA, ACAKHM and CSO algorithms [23, 24, 25]. The benchmark datasets are Iris, Wine, Glass, Breast cancer Wisconsin and CMC. These datasets are freely available on UCI repository and the characteristics of these datasets are summarized in Table 1. Mat lab 2010a environment is used to implement the proposed CSOKHM algorithm and executed on corei5 processor with 4 GB RAM. The result of proposed algorithm has taken an average of 10 simulations and the p values of performance function also play a vital role to obtain the results. The proposed CSOKHM is also tested on different p values (2.5, 3 and 3.5). The parameters setting for the proposed method is shown in Table 2.

5.1 Datasets

Synthetic dataset 1(Total instances=300, attributes=2, classes=3): It is a synthetic dataset, generated in mat lab to validate the proposed algorithm. This dataset includes 300 instances with two attributes and three classes. The data instances are generated using independent bivariate normal distribution. The classes in dataset are disseminated using μ and $\sum$where μ is the mean vector and $\sum$ is the covariance matrix. The Figure 2 depicts the synthetic dataset1 and detail description of generated dataset is given below:

$$Number\ of\ Instances(N) = \left(\mu = \begin{pmatrix}\mu_{i\,1}\\ \mu_{i\,2}\end{pmatrix}, \sum = \begin{bmatrix}0.4 & 0.04\\ 0.04 & 0.04\end{bmatrix}\right), \qquad i = 1, 2, 3 \ldots\ldots.. n$$

$$\mu_{11} = \mu_{12} = -2, \qquad \mu_{21} =, \mu_{22} = 2, \qquad \mu_{32} =, \mu_{32} = 6$$

i. Synthetic dataset 2(Total instances=300, attributes=3, classes=3): Synthetic dataset 2 includes 300 instances with three attributes and three classes. The Figure 3 represents the synthetic dataset 2 and every attribute of synthetic dataset 2 is disseminated by uniform distribution as given below

Class1 ~ Uniform (10, 25), Class2 ~ Uniform (25, 40), Class3 ~ Uniform (40, 55)

ii. Iris dataset (Total instances=150, attributes=4, classes=3): Iris dataset consist of three species of the iris flower: Iris Setosa, Iris Versicolour and Iris Virginica, The dataset consists of 150 instances and three classes. In iris dataset, each species consists of 50 instances with four attributes: sepal length, sepal width, petal length, and petal width.

iii. Wine dataset (Total instance = 178, attribute = 13, classes = 3): This dataset contains the chemical analysis of wine in the same region of Italy but three different cultivators. The dataset consists of total 178 instances and three classes with 13 attributes. The attributes of the dataset are alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and proline.

iv. Glass (Total instances = 214, attributes = 9, classes = 6): This dataset contains the information about six different types of glass. The dataset consists of total 214 instances with 10 attributes and 7 classes. The attributes of dataset are Id number, refractive index, sodium, magnesium, aluminium, silicon, potassium, calcium, barium, and iron.

v. Breast cancer wisconsim (Total instances = 683, attributes = 9, classes = 2): This dataset describes the characteristics of cell nuclei present in the image of breast mass. The dataset consists of 683 instances with 9 attributes and 2 classes i.e. malignant and benign. The attributes of dataset are clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Maligant class consists of 444 instances while benign consists of 239 instances.

vi. Contraceptive method choice (Total attributes = 1473, attributes = 9, classes = 3): CMC dataset is a subset of the National Indonesia Contraceptive Prevalence Survey that was conducted in 1987. The instances of dataset are married women who were either pregnant (but did not know about pregnancy) or not pregnant. The dataset consists of 1473 instances with 9 attribute and 3 classes. The dataset is divides into three classes i.e., no use, long term method and short term method classes and each class contains 629, 334 and 510 instances respectively. The attributes are Age, Wife's education, Husband's education, Number of children ever born, Wife's religion, Wife's now working, Husband's occupation, Standard-of-living index and Media exposure.

5.2 Parameters

i. K Harmonic Mean (KHM (X, C)): The quality of clusters is directly proportional to the distance function; smaller the sum of distances higher the quality of cluster and vice versa. Hence, the harmonic mean is defined as harmonic average of the sum of all data instance from a data instance to all cluster centers. Equation 1 is used to calculate the harmonic mean in a dataset. Therefore, minimum of harmonic average means higher the quality of clustering.

ii. F-measure: F-measure is the combination of recall and precision from an information retrieval system [26, 27]. Hence, the f-measure is defined as the weighted harmonic mean of recall and precision. To compute the value of f –measure, every cluster is represented as a result of query while every class is the preferred set of credentials for query. So, if each cluster j consists a set of n_j instances as a result for a query and each class i consists of a set of n_i instances require for a query then n_{ij} provides the numbers of instances of class i within cluster center j. The recall and precision, for each cluster j and class i is given as

$$Recall\left(r(i,j)\right) = \frac{n_{i,j}}{n_i} \quad and \quad Precision\left(p(i,j)\right) = \frac{n_{i,j}}{n_j} \tag{8}$$

The value of F-measure (F (i, j)) is computed as

$$F(i,j) = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{9}$$

Finally, the value of F-measure of a given clustering algorithm that consist of n number of data instances is given as

$$F(i,j) = \sum_{i=1}^{n} \frac{n_i}{n} . \max_1 \left(F(i,j)\right) \tag{10}$$

iii. Processing Time (Run Time): Processing time of an algorithm is defined as the amount of time taken for execution of an algorithm.

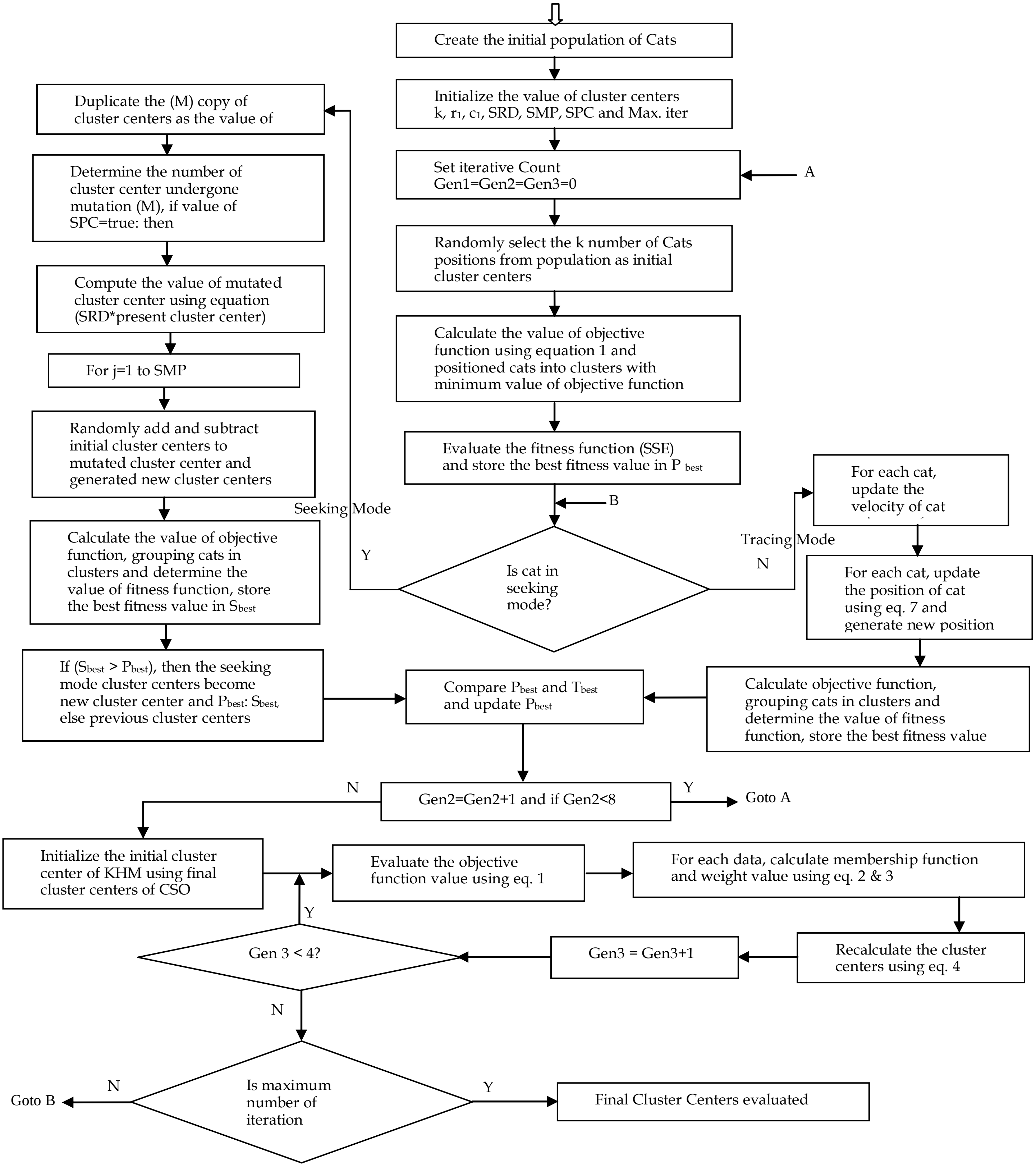

Create the initial population of Cats
Initialize the value of cluster centers k, r_1, c_1, SRD, SMP, SPC and Max. iter
Set iterative Count Gen1=Gen2=Gen3=0
A
Randomly select the k number of Cats positions from population as initial cluster centers
Calculate the value of objective function using equation 1 and positioned cats into clusters with minimum value of objective function
Evaluate the fitness function (SSE) and store the best fitness value in P_{best}
B
Is cat in seeking mode?
Seeking Mode
Y
Tracing Mode
N
Duplicate the (M) copy of cluster centers as the value of
Determine the number of cluster center undergone mutation (M), if value of SPC=true: then
Compute the value of mutated cluster center using equation (SRD*present cluster center)
For j=1 to SMP
Randomly add and subtract initial cluster centers to mutated cluster center and generated new cluster centers
Calculate the value of objective function, grouping cats in clusters and determine the value of fitness function, store the best fitness value in S_{best}
If (S_{best} > P_{best}), then the seeking mode cluster centers become new cluster center and P_{best}: $S_{best,}$ else previous cluster centers
For each cat, update the velocity of cat
For each cat, update the position of cat using eq. 7 and generate new position
Calculate objective function, grouping cats in clusters and determine the value of fitness function, store the best fitness value
Compare P_{best} and T_{best} and update P_{best}
N
Gen2=Gen2+1 and if Gen2<8
Y
Goto A
Initialize the initial cluster center of KHM using final cluster centers of CSO
Evaluate the objective function value using eq. 1
For each data, calculate membership function and weight value using eq. 2 & 3
Recalculate the cluster centers using eq. 4
Gen3 = Gen3+1
Y
Gen 3 < 4?
N
N
Goto B
Is maximum number of iteration
Y
Final Cluster Centers evaluated

Figure 1: Flowchart of proposed CSOKHM

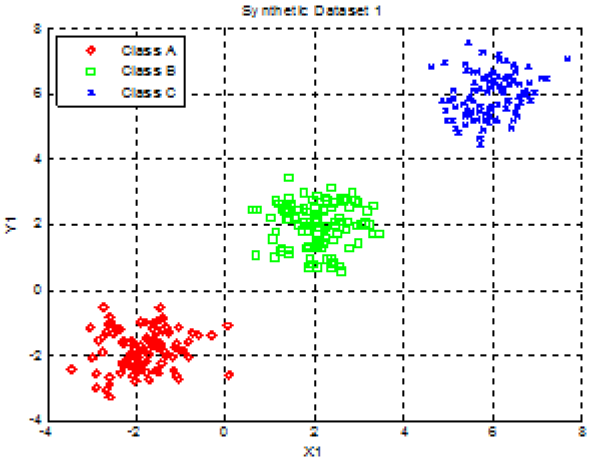


Figure 2: Represents the cluster centers in Synthetic dataset1.

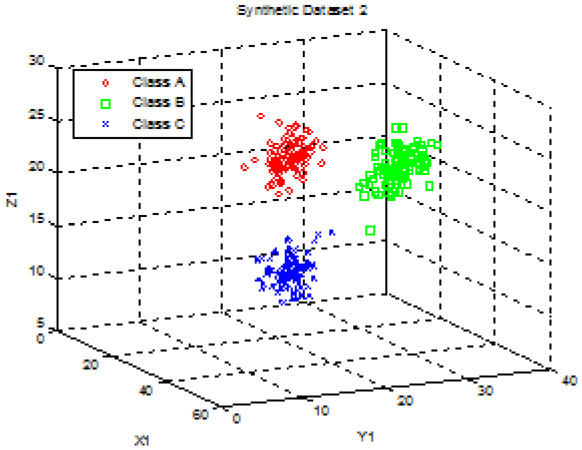
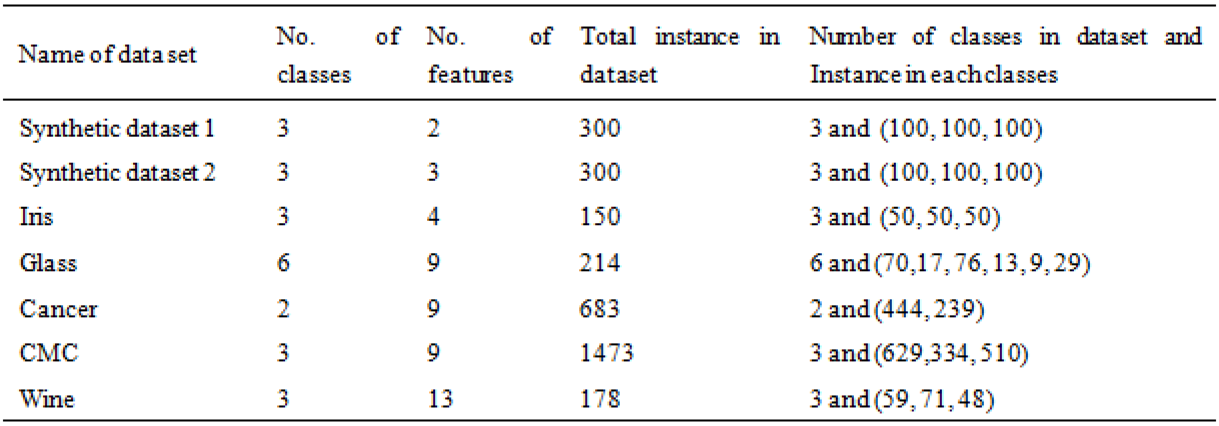


Figure 3: Represents the cluster centers in Synthetic dataset2

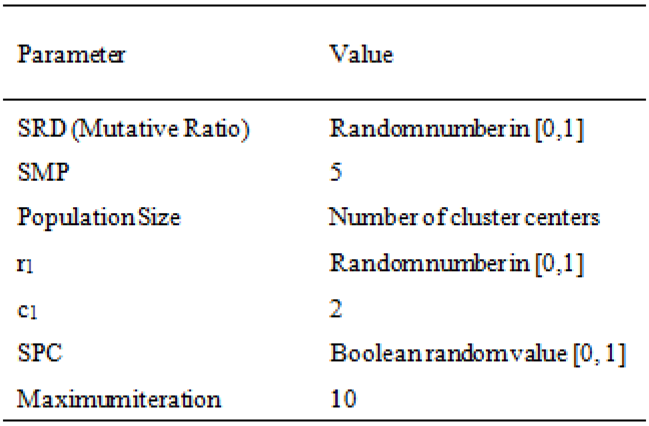
From the analysis of Tables 3, 4 and 5, it is concluded that average KHM (X, C) and f-measure of CSOKHM is much better than other methods and the KHM exhibits the poor performance among iris, wine, cancer, glass and CMC datasets while the performances of all algorithms are almost same with synthetic dataset 1 and 2 except runtime parameter. By analyzing, the runtime parameter of these methods, it is noted that KHM method requires minimum runtime than others but this algorithm run in it's inherit local optima problem. On the other side, it is seen that among all the methods being compared, the ACA method takes more time with all of datasets among all of methods From the runtime analysis of PSOKHM, ACAKHM, GSAKHM and CSOKHM, it is observed that GSAKHM algorithm requires less time for all of datasets than PSOKHM, ACAKHM and CSOKHM while CSOKHM takes more time for synthetic 1, wine and glass datasets (when p value is 2.5) and wine and glass datasets (when p value is 3.0), glass and CMC (when p value is 3.5); PSOKHM takes more time with CMC and cancer datasets (when p value is 2.5) and cancer (when p value is 3.5); ACAKHM takes more time for synthetic 2 and iris (when p value is 2.5), synthetic 1&2, iris, cancer and CMC (when p value is 3.0) and synthetic 1&2, iris and wine (when p value is 3.5). From the above analysis, it is concluded that proposed CSOKHM method provides better runtime results when p value is larger and also noticed that the CSOKHM method provides better result with complex data using all of three parameters.

Table 1: Characteristics of datasets



Name of data set	No. of classes	No. of features	Total instance in dataset	Number of classes in dataset and Instance in each classes
Synthetic dataset 1	3	2	300	3 and (100, 100, 100)
Synthetic dataset 2	3	3	300	3 and (100, 100, 100)
Iris	3	4	150	3 and (50, 50, 50)
Glass	6	9	214	6 and (70,17, 76, 13, 9, 29)
Cancer	2	9	683	2 and (444, 239)
CMC	3	9	1473	3 and (629,334, 510)
Wine	3	13	178	3 and (59, 71, 48)

Table 2: Parameters of CSOKHM



Parameter	Value
SRD (Mutative Ratio)	Randomnumber in [0,1]
SMP	5
Population Size	Number of cluster centers
r_1	Randomnumber in [0,1]
c_1	2
SPC	Booleanrandomvalue [0, 1]
Maximumiteration	10

Table 3: Shows the comparative result of KHM, PSO, ACA, CSO, PSOKHM, ACAKHM, GSAKHM and CSOKHM clustering algorithms with seven dataset (P value 2.5). Three parameters KHM (X, C), F-Measure and Run Time are used to evaluate the quality of clusters using mean and standard deviation (in brackets).

	KHM	PSO	PSOKHM	ACA	ACAKHM	GSA	GSAKHM	CSO	CSOKHM
Synthetic 1									
KHM(X,C)	703.863 (0.011)	703.514 (0.026)	703.502 (0.050)	701.750 (0.003)	703.511 (0.006)	703.566 (0.0326)	703.50 4 (0.014)	703.614 (0.003)	703.498 (0.004)
F-Measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.2)	1.00 (0.00)	1.000 (0.004)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Processing Time	0.105 (0.003)	1.628 (0.018)	1.911 (0.012)	1.756 (0.018)	1.960 (0.02)	1.923 (0.008)	1.779 (0.021)	1.736 (0.012)	1.935 (0.006)
Synthetic 2									
KHM(X,C)	111,882 (0)	1,910,796 (915,412)	111,723 (14)	1,912,465 (6174)	111,793 (89)	1,911,018 (891,434)	111,719 (19)	1,909,896 (856,728)	111,716 (21)
F-Measure	1.000 (0.000)	0.668 (0.078)	1.000 (0.000)	0.664 (0.063)	1.00 (0.00)	0.667 (0.026)	1.000 (0.000)	0.672 (0.038)	1.000 (0.000)

Processing Time	0.22 3 (0.008)	3.650 (0.031)	2.859 (0.000)	4.61 (0.011)	3.08 (0.029)	3.442 (0.012)	2.461 (0.001)	3.489 (0.056)	2.951 (0.017)
Iris									
KHM(X,C)	149.333 (0.000)	230.340 (98.180)	149.058 (0.074)	189.9(250)	149.34(19.5)	234.14 (43.18)	149.058 (0.000)	216.12 (18.662)	148.946 (9.873)
F-Measure	0.750 (0.000)	0.711 (0.062)	0.753 (0.005)	0.33(0.16)	0.79 (0.01)	0.709 (0.016)	0.763 (0.000)	0.723 (0.037)	0.778 (0.063)
Processing Time	0.192 (0.008)	3.117 (0.020)	1.842 (0.005)	4.05(0.002)	3.59 (0.017)	3.018 (0.022)	1.577 (0.002)	3.016 (0.043)	2.237 (0.014)
Glass									
KHM(X,C)	1203.554 (16.231)	9551.095 (1933.211)	1196.798 (0.439)	601 (663)	572.9(0.00)	9458.095 (1172.67)	1180.756 (0.134)	9467.562 (1630.823)	1238.213 (0.452)
F-Measure	0.421 (0.011)	0.387 (0.044)	0.424 (0.003)	0.28(0.3)	0.40(0.00)	0.387 (0.044)	0.454 (0.000)	0.407 (0.011)	0.462 (0.018)
Processing Time	4.064 (0.010)	44.249 (0.431)	17.669 (0.018)	45.35 (0.008)	16.89 (0.025)	44.249 (0.431)	15.910 (0.010)	46.126 (0.021)	22.013 (0.019)
Cancer									
KHM(X,C)	60,189 (0)	60,244 (563)	59,844 (22)	61269.86 (347)	59864.78 (923.2)	60,073 (287)	59844 (0)	60, 136 (324)	59,844 (19)
F-Measure	0.829 (0.000)	0.819 (0.005)	0.829 (0.000)	0.28(0.12)	0.53(0.02)	0.819 (0.036)	0.829 (0.000)	0.823 (0.007)	0.829 (0.002)
Processing Time	2.017 (0.009)	16.046 (0.138)	9.525 (0.013)	17.572(0.026)	11.28(0.050)	15.128 (0.204)	7.509 (0.007)	17.128 (0.026)	8.012 (0.006)
CMC									
KHM(X,C)	96,520 (0)	115,096 (33,014)	96,193 (25)	104169.86 (24447.1)	967164.78 (23)	114,846 (26,564)	96193 (52)	114,362 (17,348)	96, 113 (28)
F-Measure	0.335 (0.000)	0.298 (0.019)	0.333 (0.002)	0.29(0.08)	0.53(0.054)	0.302 (0.037)	0.488 (0.000)	0.320 (0.022)	0.336 (0.004)
Processing Time	8.639 (0.009)	54.163 (0.578)	39.825 (0.072)	56.78(0.023)	33.42(0.036)	49.245 (0.326)	31.563 (0.012)	47.835 (0.034)	36.141 (0.072)
Wine									
KHM(X,C)	18,386,505 (0)	19,795,542 (2,007,722)	18,386,285 (5)	19,458,595 (447,172)	18,386,294 (32)	19,794,573 (2,126,573)	18,386,285 (28)	18,473,397 (405,216)	18,386, 318 (24)
F-Measure	0.516 (0.000)	0.512 (0.020)	0.516 (0.000)	0.514 (0.020)	0.519 (0.000)	0.512 (0.314)	0.519 (0.000)	0.518 (0.016)	0.526 (0.003)
Processing Time	2.059 (0.010)	35.642 (0.282)	6.539 (0.008)	37.682 (0.062)	6.248 (0.016)	36.832 (0.098)	5.628 (0.004)	43.272 (0.076)	7.159 (0.012)

Table 4: Shows the comparative result of KHM, PSO, ACA, CSO, PSOKHM, ACAKHM, GSAKHM and CSOKHM clustering algorithms with seven dataset (P value 3). Three parameters KHM (X, C), F-Measure and Run Time are used to evaluate the quality of clusters using mean and standard deviation (in brackets).

	KHM	PSO	PSOKHM	ACA	ACAKHM	GSA	GSAKHM	CSO	CSOKHM
Synthetic 1									
KHM(X,C)	742.116	741.682	741.458	741.699	741.467	741.688	741.453	741.663	741.441

	(0.004)	(0.076)	(0.002)	(0.056)	(0.017)	(0.012)	(0.000)	(0.013)	(0.000)
F-Measure	1.000 (0.000)	1.0000 (0.000)	1.000 (0.000)	1.0000 (0.000)	1.000 (0.000)	1.0000 (0.000)	1.000 (0.000)	1.0000 (0.000)	1.000 (0.000)
Processing Time	0.001 (0.003)	1.633 (0.012)	1.921 (0.007)	1.796 (0.021)	1.929 (0.007)	1.716 (0.008)	1.789 (0.002)	1.746 (0.006)	1.834 (0.02)
Synthetic 2									
KHM(X,C)	278,758 (0)	8,675,830 (6,626,165)	278,541 (33)	8,675,172 (6,625,756)	278,545 (11)	8,675,812 (6,625,896)	278,541 (11)	8,674,735 (6,625,865)	278,537 (16)
F-Measure	1.000 (0.000)	0.681 (0.093)	1.000 (0.000)	0.682 (0.045)	1.000 (0.000)	0.681 (0.256)	1.000 (0.000)	0.683 (0.093)	1.000 (0.000)
Processing Time	0.220 (0.005)	3.575 (0.030)	2.844 (0.010)	5.42 (0.024)	4.118 (0.039)	3.542 (0.046)	2.524 (0.005)	3.486 (0.022)	2.654 (0.06)
Iris									
KHM(X,C)	126.517 (0.000)	147.217 (22.896)	125.951 (0.052)	147.378 (26)	126.216 (0.078)	147.209 (20.634)	125.951 (0.000)	146.92 (21.453)	125.736 (0)
F-Measure	0.744 (0.000)	0.740 (0.025)	0.744 (0.000)	0.746 (0.005)	0.746 (0.03)	0.743 (0.018)	0.751 (0.000)	742 (0.011)	756 (0)
Processing Time	0.190 (0.007)	3.096 (0.010)	2.796 (0.010)	4.27 (0.003)	3.70 (0.021)	2.998 (0.024)	1.650 (0.004)	3.213 (0.013)	2.865 (0.008)
Glass									
KHM(X,C)	1535.198 (0.000)	18191.700 (1870.044)	1442.847 (35.871)	18298.62 (1540)	1448.366 (86)	18246.0119 (956.238)	1400.950 (0.630)	16167.562 (1570.782)	1389.278 (0.563)
F-Measure	0.422 (0.000)	0.378 (0.030)	0.427 (0.003)	0.371 (0.024)	0.422 (0.016)	0.378 (0.030)	0.442 (0.000)	0.423 (0.017)	0.448 (0.006)
Processing Time	4.042 (0.007)	43.594 (0.338)	17.609 (0.015)	44.67 (0.010)	16.28 (0.037)	45.385 (0.428)	15.958 (0.001)	47.246 (0.029)	23.083 (0.006)
Cancer									
KHM(X,C)	119,458 (0)	119,333 (3770)	117,418 (237)	120,104 (3580)	117,468 (196)	118,412 (1236)	117,418 (55)	118, 936 (378)	117,418 (46)
F-Measure	0.834 (0.000)	0.817 (0.033)	0.834 (0.000)	0.807 (0.092)	0.836 (0.013)	0.826 (0.202)	0.847 (0.000)	829 (0.015)	853 (0.005)
Processing Time	2.027 (0.007)	16.150 (0.144)	9.594 (0.023)	13.926 (0.012)	12.53 (0.015)	15.638 (0.372)	7.91 (0.002)	16.135 (0.096)	8.091 (0.036)
CMC									
KHM(X,C)	187,525 (0)	205,548 (60,798)	186,722 (111)	208,278 (55,768)	186,856 (42)	204,986 (61,369)	186,722 (94)	203,474 (55,989)	186, 713 (85)
F-Measure	0.303 (0.000)	0.250 (0.028)	0.303 (0.000)	0.256 (0.008)	0.296 (0.033)	0.267 (0.143)	0.472 (0.000)	0.291 (0.042)	0.493 (0.007)
Processing Time	8.627 (0.009)	54.895 (0.933)	39.485 (0.056)	54.242 (0.014)	39.576 (0.029)	56.559 (0.191)	32.107 (0.034)	57.835 (0.034)	36.141 (0.072)
Wine									
KHM(X,C)	298,230,848 (24,270,951)	276,508,278 (23,807,035)	252,522,504 (766)	276,506,876 (21,670,255)	252,526,114 (274)	276,506,778 (23,7867,414)	252,522,000 (0)	273,473,397 (19,405,216)	252, 492,116 (324)
F-Measure	0.538	0.519	0.553	0.519	0.551	0.521	0.553	0.526	0.554

	(0.007)	(0.021)	(0.000)	(0.021)	(0.006)	(0.013)	(0.000)	(0.014)	(0.003)
Processing Time	2.084 (0.010)	35.284 (0.531)	6.598 (0.008)	35.46 (0.031)	6 .16 (0.026)	34.756 (0.464)	5.710 (0.001)	38.972 (0.476)	7.857 (0.005)

Table 5: Shows the comparative result of KHM, PSO, ACA, CSO, PSOKHM, ACAKHM, GSAKHM and CSOKHM clustering algorithms with seven dataset (P value 3.5). Three parameters KHM (X, C), F-Measure and Run Time are used to evaluate the quality of clusters using mean and standard deviation (in brackets).

	KHM	PSO	PSOKHM	ACA	ACAKHM	GSA	GSAKHM	CSO	CSOKHM
Synthetic 1									
KHM(X,C)	807.548 (0.016)	806.811 (0.079)	806.619 (0.014)	807.742 (0.08)	807.514 (0.06)	806.779 (0.027)	806.613 (0.012)	806.708 (0.037)	806.532 (0.072)
F-Measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.0000)	1.00 (0.00)	1.000 (0.000)	1.000 (0.000)	1.000 .000)	1.000 (0.000)
Processing Time	0.106 (0.006)	1.628 (0.006)	1.921 (0.007)	5.39 (0.015)	3.53 (0.017)	1.616 (0.006)	1.766 (0.001)	1.725 (0.016)	1.856 (0.005)
Synthetic 2									
KHM(X,C)	697,215 (0.000)	80,729,943 (33,400,802)	696,349 (78)	80,730,423 (33, 350, 517)	697,105 (0.00)	80,726,839 (33,401,426)	696,281 (1)	80,727,883 (33,378,953)	696,226 (26)
F-Measure	1.000 (0.000)	0.660 (0.081)	1.000 (0.000)	0.645 (0.415)	1.00(0.00)	0.662 (0.054)	1.000 (0.000)	0.679 (0.076)	1.000 (0.000)
Processing Time	0.220 (0.005)	3.601 (0.025)	2.842 (0.005)	6.51(0.015)	4.19(0.027)	3.624 (0.037)	2.471 (0.001)	3.896 (0.049)	2.814 (0.034)
Iris									
KHM(X,C)	113.413 (0.085)	255.763 (117.388)	110.004 (0.260)	2089.38 (1619)	112.466 (1.2)	242.566 (112)	110.004 (0.002)	228.534 (87.216)	110.004 (0.026)
F-Measure	0.770 (0.024)	0.660 (0.057)	0.762 (0.004)	0.643 (0.09)	0.80 (0.07)	0.672 (0.038)	0.766 (0.000)	0.693 (0.024)	0.767 (0.005)
Runtime	0.194 (0.008)	3.078 (0.013)	1.873 (0.005)	4.27 (0.003)	3.70 (0.021)	3.107 (0.029)	1.587 (0.004)	3.458 (0.002)	1.793 (0.006)
Glass									
KHM(X,C)	1871.812 (0.000)	32933.349 (1398.602)	1857.152 (4.937)	76125 (1415)	1871.811617 (0.00)	32933.349 (1398.602)	1857.152 (0.035)	31786.789 (264.534)	1857.152 (0.456)
F-Measure	0.396 (0.000)	0.373 (0.020)	0.396 (0.000)	0.27 (0.3)	0.40 (0.00)	0.386 (0.046)	0.420 (0.000)	0.382 (0.015)	0.416 (0.000)
Processing Time	4.056 (0.008)	43.350 (0.332)	17.651(0.013)	41.067 (0.010)	16.28 (0.037)	42.218 (0.178)	15.799 (0.003)	41.108 (0.246)	17.867 (0.002)
Cancer									
KHM(X,C)	243,440 (0)	240,634 (8842)	235,441 (696)	241682 (6,327)	236341 (125.78)	240,484 (6032)	236,125 (15)	240,118 (1040)	235,965 (45)
F-Measure	0.832 (0.000)	0.820 (0.046)	0.835 (0.003)	0.824 (0.46)	0.876 (0.062)	0.823 (0.028)	0.862 (0.000)	0.826(0.023)	0.889 (0.012)
Runtime	2.072 (0.008)	42.097 (0.095)	39.859 (0.015)	45.26 (0.041)	36.53 (0.027)	41.513 (0.162)	31.521 (0.009)	41.679 (0.012)	36.521 (0.009)

CMC									
KHM(X,C)	381,444 (0)	423,562 (43,932)	379,678 (247)	424,744 (36,214)	380,462 (578.98)	423,096 (39,973)	380,183 (16)	422,846 (40,565)	380,069 (47)
F-Measure	0.332 (0.000)	0.298 (0.016)	0.332 (0.000)	0.311 (0.67)	0.514 (0.062)	0.301 (0.126)	0.506 (0.000)	0.309 (0.008)	0.521 (0.004)
Processing Time	8.528 (0.012)	49.881 (0.256)	32.7017 (0.250)	46.236 (0.041)	36.268 (0.027)	49.035 (0.086)	31.521 (0.009)	49.485 (0.178)	38.456 (0.012)
Wine									
KHM(X,C)	8,568,319,639 (2075)	3,637,575,95 (202,759,448)	3,546,930,579 (1,214,985)	7,285,431,684 (2,784,324)	3,549,156,713 (208.78)	3,637,563,41 (202,747,531)	3,540,920,000 (232)	3,637,564,74 (202,754,594)	3,540,918,728 (476)
F-Measure	0.502 (0.000)	0.530 (0.039)	0.535 (0.004)	0.519 (0.43)	0.534 (0.06)	0.530 (0.119)	0.536 (0.000)	0.531 (0.002)	0.542 (0.008)
Processing Time	2.040 (0.008)	35.072 (0.385)	6.508 (0.017)	35.846 (0.031)	7.57 (0.050)	35.512 (0.288)	5.536 (0.001)	35.102 (0.005)	6.236 (0.007)

6. Conclusions

In this paper, a new hybrid CSOKHM algorithm is proposed with the combination of cat swarm optimization (CSO) and k-harmonic means (KM). The performance of the proposed algorithm is investigated on two synthetic and five benchmark datasets and compared with KHM, PSO, PSOKHM, ACA, ACAKHM, GSA and GSAKHM. The investigational results of CSOKHM prove that it is an effective and more competent algorithm than other existing algorithms for clustering problem.In this paper, harmonic average is used as objective function for CSOKHM instead of Euclidean distance. With the same objective function, the CSO method requires more time as well as convergence while KHM is stuck in local optima. Moreover, the proposed CSOKHM algorithm not only improves the convergence speed of CSO but also escapes the KHM to run in local optima. But, it is observed that the algorithm takes more time for its execution. Hence,

7. References

[1]. Garey, M. R., D. Johnson, and Hans Witsenhausen. "The complexity of the generalized Lloyd-max problem (corresp.)." Information Theory, IEEE Transactions on 28, no. 2, pp 255-256, 1982.

[2]. MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 281-297. 1967.

[3]. Pena, José Manuel, Jose Antonio Lozano, and Pedro Larranaga. "An empirical comparison of four initialization methods for the< i> K</i>-Means algorithm."Pattern recognition letters 20, vol. 10, pp 1027-1040, 1999.

[4]. Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern Recognition Letters 31, no. 8, pp 651-666, 2010.

[5]. Zhang, Bin, Meichun Hsu, and Umeshwar Dayal. "K-harmonic means-a data clustering algorithm." Hewlett-Packard Labs Technical Report HPL-1999-124, 1999.

[6]. Dervis Karaboga and Celal Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Applied Soft Computing 11, pp 652–657, 2011.

[7]. Shelokar, P. S., Valadi K. Jayaraman, and Bhaskar D. Kulkarni. "An ant colony approach for clustering." Analytica Chimica Acta 509, no. 2, pp 187-195, 2004.

[8]. Murthy, Chivukula A., and Nirmalya Chowdhury. "In search of optimal clusters using genetic algorithms." Pattern Recognition Letters 17, no. 8 (1996): 825-832.

[9]. Chen, Ching-Yi, and Fun Ye. "Particle swarm optimization algorithm and its application to clustering analysis." In Networking, Sensing and Control, 2004 IEEE International Conference on, vol. 2, pp. 789-794. IEEE, 2004.

[10]. Santosa, Budi, and Mirsa Kencana Ningrum. "Cat swarm optimization for clustering." In International Conference on Soft Computing and Pattern Recognition (SOCPAR'09), pp. 54-59, 2009.

[11]. Hatamlou, Abdolreza. "Black hole: A new heuristic optimization approach for data clustering." Information Sciences 222, pp 175-184, 2013.

[12]. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. “GSA: a gravitational search algorithm”, Information sciences, 179(13), pp 2232-2248, 2009.

[13]. Zhang, Jun, Henry Shu-Hung Chung, and Wai-Lun Lo. "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms", IEEE Transactions on Evolutionary Computation vol. 11, no. 3, pp 326-335, 2007.

[14]. Mullen, Robert J., Dorothy Monekosso, Sarah Barman, and Paolo Remagnino. "A review of ant algorithms." Expert Systems with Applications 36, no. 6, pp 9608-9617, 2009.

[15]. Rana, Sandeep, Sanjay Jasola, and Rajesh Kumar. "A review on particle swarm optimization algorithms and their applications to data clustering." Artificial Intelligence Review 35, no. 3, pp 211-222, 2011.

[16]. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Global Optim. 39 (3), pp 459–471, 2007.

[17]. S. Sarafrazi, H. Nezamabadi-pour and S. Saryazdi, Disruption: A new operator in gravitational search algorithm, Scientia Iranica D, 18 (3), 539–548, 2011.

[18]. Chu, Shu-Chuan, Pei-Wei Tsai, and Jeng-Shyang Pan. "Cat swarm optimization." In PRICAI 2006: Trends in Artificial Intelligence, pp. 854-858. Springer Berlin Heidelberg, 2006.

[19]. Panda, Ganapati, Pyari Mohan Pradhan, and Babita Majhi. "IIR system identification using cat swarm optimization." Expert Systems with Applications 38, no. 10, pp 12671-12683, 2011.

[20]. Tsai, Pei-Wei, Jeng-shyang Pan, Shyi-Ming Chen, Bin-Yih Liao, and Szu-Ping Hao. "Parallel cat swarm optimization." In Machine Learning and Cybernetics, 2008 International Conference on, vol. 6, pp. 3328-3333. IEEE, 2008.

[21]. Pradhan, Pyari Mohan, and Ganapati Panda. "Solving multi objective problems using cat swarm optimization." Expert Systems with Applications 39, no. 3, pp 2956-2964, 2012.

[22]. Zhang, Bin. "Generalized k-harmonic means." Hewlett-Packard Laboratories Technical Report (2000).

[23]. Yang, Fengqin, Tieli Sun, and Changhai Zhang. "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization." Expert Systems with Applications 36, no. 6, pp 9847-9852, 2009.

[24]. Yin, Minghao, Yanmei Hu, Fengqin Yang, Xiangtao Li, and Wenxiang Gu. "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering." Expert Systems with Applications 38, no. 8, pp 9319-9324, 2011.

[25]. Jiang, Hua, Shenghe Yi, Jing Li, Fengqin Yang, and Xin Hu. "Ant clustering algorithm with< i> K</i>-harmonic means clustering." Expert Systems with Applications 37, no. 12, 8679-8684, 2010.

[26]. Dalli, Angelo. "Adaptation of the F-measure to cluster based lexicon quality evaluation." In Proceedings of the EACL 2003 pp. 51-56. Association for Computational Linguistics, 2003.

[27]. Handl, J., Knowles, J., & Dorigo, M. (2003). On the performance of ant-based clustering. Design and Application of Hybrid Intelligent Systems. Frontiers in Artificial Intelligence and Applications, 104, pp 204–213, 2003.