A Variant of Genetic Algorithm Based Categorical Data Clustering for Compact Clusters and an Experimental Study on Soybean Data for Local and Global Optimal Solutions

Abha Sharma
Maulana Azad National Institute of Technology,
Bhopal, India

R. S. Thakur
Maulana Azad National Institute of Technology,
Bhopal, India

***Abstract*—Almost all partitioning clustering algorithms getting stuck to the local optimal solutions. Using Genetic algorithms (GA) the results can be find globally optimal. This piece of work offers and investigates a new variant of the Genetic algorithm (GA) based *k*-Modes clustering algorithm for categorical data. A statistical analysis have been done on the popular categorical dataset which shows the user specified cluster centres stuck at local optimal solution in *k*-Modes algorithm even in all the higher iterations and the proposed algorithm overcome this problem of local optima. To the best of our knowledge, such comparison has been reported here for the first time in case of categorical data. The obtained results, shows that the proposed algorithm is better over the conventional *k*-Modes algorithm in terms of optimal solutions and within cluster variation measure.**

Keywords*—Clustering; Categorical data; k-Modes; Genetic Algorithm

I. INTRODUCTION

There is a growing requirement for the way to extract knowledge from the data [1]. Clustering is a descriptive task which partition the dataset based on the predefined similarity measure [2]. Clustering techniques have been widely used in machine learning, pattern recognition, medical etc. Number of clustering algorithms have been proposed for different requirements and nature of the data [3]. Partition based clustering (k-Modes and its initialisation methods) [4], hierarchical clustering (HIERDENC) [5] model-based clustering (EAST algorithm) [6], density-based clustering [7], graph-based clustering, and grid-based clustering are some basic clustering algorithms with their advantages and disadvantages.

It is hard to discover the distance measure between two categorical data objects, greater the distance between the clusters more separated will be the clusters [8]. One of the well-known clustering for categorical data is *k*-Modes algorithm for large datasets. The traditional way to treat categorical data is binary but does not do justice to the large value difference such as for the very low and very high the difference is same.

The major issue in partition clustering is to initialize the cluster centres, since it has a direct influence on the construction of ultimate clusters. This paper focus on the better partitions of all real world categorical datasets on the lowest cost using GA in less space and time.

GA is proposed by Holland [9] and can apply to many optimization problems. Due to the cluster centre initialization problem which affects the proper clustering of data, GA has been used to convert the local optimal solution into global optimal solution in many GA based clustering algorithm for numeric as well as categorical data in the literature [10]. This paper calculates Total Within Cluster Variation (TWCV), time and conversion of local optima to global optima. Fig 1. shows the various operators used in GA based categorical data clustering found in literature.

This paper is organized as follows: Section II presents Literature Review: Section III presents Background: Section IV presents proposed method: Section V shows the Experimental details where we compare basic *k*-Modes algorithm with propose algorithms: Section VI concludes the paper: Section VII tries to put the future work.

II. LITERATURE REVIEW

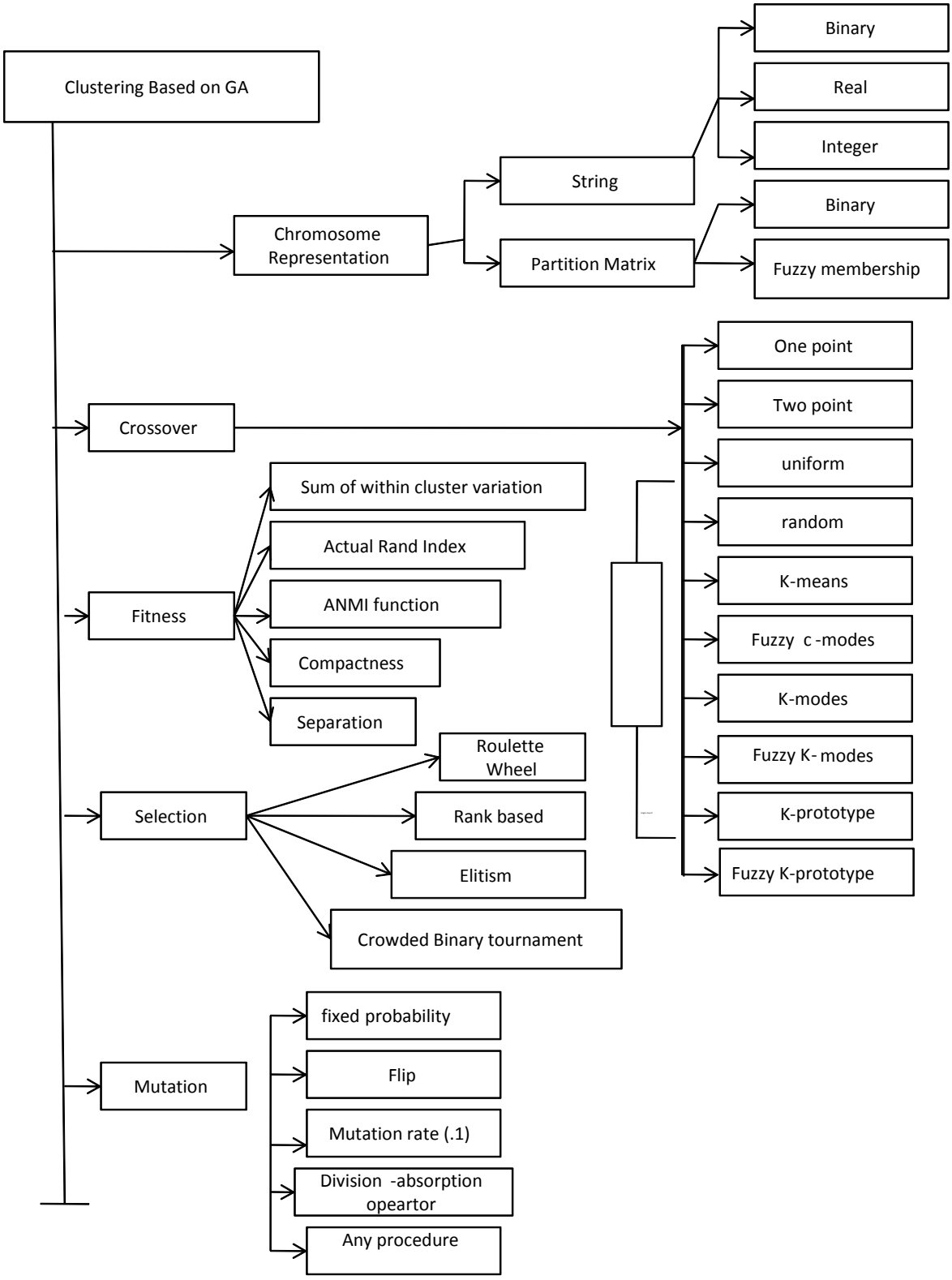

Fig. 1. Many operators used in GA based clustering according to Literature

TABLE I. COMPARISON OF ALGORITHMS

Operators	**NSGA-FMC [10]**	**MOGA**	**G-AMNI [13]**	**Improved G-AMNI [11]**	**AGCUK [12]**	**GA and Simulated annealing based CDC**	**A Genetic *k*-Modes Algorithm [15]**
Chromosome representation	Fuzzy membership matrix kxN matrix. where n=number of data objects K=number of cluster	K x A matrix K=number of clusters A= categorical attributes Chromosomes=set of clustering centres , Comparatively small	-	K x N	The length of the chromosome is K x m, where K =number of clusters and m =number of attributes	Kx N where *K* is the number of clusters and *n* is the number of points.	NxK partition matrix
Population initialization	Code with random numbers	*K* random objects of the categorical dataset of *P(population size)* chromosomes in the population	randomly selected partitions of objects	-		randomly selected partitions of objects	randomly
Chromosome selection	Roulette wheel strategy	Crowded binary tournament selection	Roulette wheel strategy	Roulette wheel strategy		-	-
Fitness of chromosome	Rank based evaluation function	Separation and compactness function	ANMI Function $\Phi^{(ANMI)}=1/r\sum \Phi^{(NMI)}(\lambda^{(q)}, \lambda)$	ANMI function	Davies–Bouldin (DB) index		
Crossover	One step fuzzy k-modes crossover operator	single-point crossover depending on crossover probability μc	Single point crossover, crossover site is selected randomly	-	-	single point crossover with a fixed crossover probability of μc	One step kmode operator
Mutation	.01 mutation	Two step mutation probability μm (a) the gene position is selected randomly (b) the categorical value of that position is replaced by another random value chosen from the corresponding categorical domain.	uniform probability	-	Division–absorption mutation Division operation: the most sparse cluster is determined Absorption operation: determine which cluster is to be merged	fixed probability μm.	The mutation operator changes an allele value depending on the distance between the cluster center and the corresponding data point.
Selection	-	-	roulette wheel strategy, if best chromosome not found use elite selection	-	Elitist operation	Propotional/ roulette wheel	-
Sorting	Elitism non-dominated sorting plus crowded tournament selection is used to evaluate the clustering solution.	-	--	-	-	-	-
Termination criteria	--	-	-	-	In general, two stopping criteria are used in genetic algorithms: based on fixed number of iteration and no further improvement in fitness value of the best individual. This work used fixed number of iteration.	Fixed number of iterations. Elitism at each generation.	-
Tested Dataset	*Soybean Zoo*	*Zoo Soybean*	*Brest cancer Vote*	*Zoo Vote*	*Breast cancer Wisconsin breast*	*Soybean Zoo*	*Mushroom Votes*

	Votes	*Breast Cancer Vote*	*Zoo Mushroom*	*Brest cance Mushroom*	*cancer*	*Tic Tac Toe*	*Zoo*
Validity Measure	Compactness, Separation, Time complexity, Actual Rand Index	-	Clustering Error	Accuracy	-	Mincowski value	Corrected Rand Index

III. BACKGROUND

In many categorical data clustering algorithms the seeds or the cluster centres are not known in advance for example *k*-Modes algorithm is a well-known and widely used clustering technique of this type. However, the major drawback of the *k*-Modes is that it often gets stuck at local minima and the result is largely dependent on the choice of the initial cluster centres.

A. K-Modes Algorithm

a) Dissimilarity measure

Let *A* and *B* be two categorical objects described by *m* categorical attributes. The dissimilarity measure can be defined by the total mismatches of the corresponding attribute categories of the two objects [4]. Formally

$$d_1(A,B) = \sum_{j=1}^{m} d(a_i,b_i) \text{-----(1)}$$

Where

$$d\chi^2(A,B) = \sum_{j=1}^{m} \frac{(n_{aj} + n_{bj})}{n_{aj} * n_{bj}} \delta(a_j,b_j) ----(3) \qquad (2)$$

where n_{aj} and n_{bj} are the number of objects in the dataset that have categories a_j and b_j for attribute j and $d\chi^2$ (X, Y) is Chi-square distance.

This paper work on dataset having frequencies of categories then the distance calculation [4] eq. (2) is used to calculate the distance.

Consider X is set of categorical objects described by categorical attributes, $C_1; C_2; : : : ; C_u$

Definition 1. Mode of X is a vector $M = [m_1, m_{2,.....}, m_n]$ that minimizes $D(X,\mathrm{M}) = \sum_{j=1}^{m} d_1(X_i,\mathrm{M})$. Where M may or may not the element of X. [4]

b) Find a mode for a set

Let $n_{tk,\,j}$ be the number of data objects having the k^{th} category $t_{k,\,j}$ in attribute C_j and the relative frequency of $t_{k,\,j}$ in X is $frC_j = t_{k,\,j}|X = \frac{n_{tk,\,j}}{n}$ [4]

Theorem 1. The function D(X, M) is minimized if $fr(C_j = m_j \mid X) \geq f_r(A_j = t_{k,\,j} \mid X)$ for $q_j \neq c_{k,\,j}$ for all j=1, 2,..,m.

c) The k-Modes algorithm

When equation (1) and (2) are used as the dissimilarity measure for categorical objects, the cost function becomes

$$P(W,M) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,l}\delta(x_{i,\,j}, m_{l,\,j}) ------(3)$$

where $w_{i,\,j} \in W$ and $M_l = [m_{l1},\ m_{l1},\ .....,\ m_{lu}] \in M$

IV. PROPOSED WORK

An attempt is made in this paper to integrate the effectiveness of the *k*-Modes algorithm for partitioning data into a number of clusters, with the capability of genetic algorithm to bring it out of this local minima. GAs are randomized search and are efficient to provide near optimal solutions of fitness function in an optimization problem.

The Local Search based approach such as *k*-Modes may get stuck at the local optimum solutions. Genetic algorithm based clustering escape from the local optimum, but it is slow and expensive to compute. The Similarity based approaches are not consistent among different inputs and can be context dependent. A small gap between k-modes and proposed GA based algorithm is the assumption of cluster centres on which the clustering is based. GA is an efficient algorithm to solve optimization problems which represented by chromosomes as string encodings and has multiples solutions. GA opts for the best fit solutions in each generation.

Increase in the string length of the chromosome, the search space in GAs increases therefore the whole process becomes more time consuming. When the number of data points and number of attributes are very large then the size of a chromosome which is equal to the number of data points multiplied by number of clusters assumed is difficult to store and manage.

In this paper, a generalized mechanism for all the categorical datasets is presented to identify and ignore the worst cluster centres in a categorical data set. Proposed work utilizes the robustness of genetic algorithm (GA) to optimize the *k*-Modes clustering algorithm that uses searching capability of GAs to determine most appropriate cluster centres which also prohibits the expensive crossover operator by using one step *k*-Modes operator. The associated cost function is defined in terms of the distances between the cluster objects and the cluster centre. This paper presents chromosomes in the form of strings (sequence of data values).

The objective of this work is to find *k* partitions that minimize the cost function and find optimal solution with some GA operators; string representation, population size, selection operator and one-step *k*-Modes algorithm in the place of the crossover operator This paper shows how the conventional k-Modes clustering algorithm may stops at locally optimal solution whereas the proposed hybridized clustering algorithm facilitate the global optimization of the underlying cost as objective function, to construct optimal partition of objects so that the within-cluster dispersion can be minimized and the between-cluster separation can be increased.

a) Encoding for categorical data:

Variant size of matrices are developed for chromosome representation in almost all GA based categorical data clustering. In this paper the chromosomes are encoded as string with $N*k$ size [14].

Example 1. Suppose N=2 and k=4 then the string representation for a chromosome is (Yellow Small Stretch Adult Purple Large Dip Child) from Lenses real world dataset. It embed the two clusters (Yellow Small Stretch Adult) and (Purple Large Dip Child). Each categorical data in the chromosome is a allele.

Consequently the updated cluster mode is (Yellow Small Stretch Adult) (Purple Large Dip Child) with the frequency based method shown in equation (2) later, the cost or within cluster variation is calculated.

b) Fitness calculation

This paper presents fitness function as the sum of within clustering variation, larger the fitness, denser the data in cluster and more separated from the other clusters. The details are described below.

Initially the clusters are formed randomly using the centres encoded in that particular chromosome, then the cluster centres encoded in the chromosome are replaced by the cluster centre (modes) of the respective clusters using frequency method. Therefore assign each point x_i=1, 2, 3,...,n_i with mode m_j such that

$$\left\|x_i - m_i\right\| \; < \; \left\|x_i - m_t\right\|, \; t = 1, \; 2, 3 \; , ...k$$

Frequency method shown in eq. (2) for attribute j where C_j is replaced by new C_i.

$$fr \; C_j * = t_{k,\,j} \mid X = \frac{n_{t_{k,\,j}}}{n}$$

Therefore the fitness is calculated using following equation

$$P(W, M) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,\,l} \delta(x_{i,\,j}, m_{l,\,j}) \quad \text{--------(4)}$$

The fitness function is defined as *f=1/P(W,M)* i.e. less the cost more fit will be the chromosome.

c) Selection

The fundamental selection method for GA based clustering algorithm is spinning the roulette wheel. In this paper after fitness calculation sort the cost of all the chromosomes in the population in the present generation, delete if the highest cost of chromosome in present generation is greater than the average of all cost of the chromosomes in the next iteration else keep that chromosome in that population.

d) Crossover process

Similar to genetic *k*-Modes algorithm, this paper also used one step *k*-Modes algorithm as the crossover operator to exchange of information between the two parent chromosomes to generate two offspring's.

e) Termination criteria

The most popular termination criteria for GA based clustering algorithms are: to run the algorithm based on user defined iterations. In the proposed algorithm iteration stops for the particular chromosome if the constant fitness value persist even before user specified iteration count.

f) Solution of the Empty cluster problem

The Empty cluster formation is the well-known problem in clustering. And the problem becomes big if the optimization techniques are used, this paper try to remove the empty cluster issue using following algorithm:

Algorithm:

If (In any generation for C_i the intermediate clusters in chromosome are found to be null or empty)

```
{
 iteration ++
if(found any empty cluster)
{
delete the chromosome & M=cost C_i
}
else
go head
else go to next iteration ()
    }
```

Example 2. Suppose N=4 and k=2 if the intermediate clusters

After first generation:

(Yellow Small Stretch Adult) (Purple Large Dip Child)

After Second generation:

(Yellow Adult Stretch Child) (Purple Large Dip Adult)

.....
.....
.....

After m^{th} generation

(Yellow Small Dip Child) ( )

After n^{th} generation

( ) ( )

After using the above algorithm the updated clusters are:

(Yellow Small Stretch Adult) (Purple Large Dip Child)

In the current implementation of GA this paper used the standard *k*-Modes algorithm for creating multiple partitions of the categorical data for global optimal solution.

B. *Flowchart of proposed GA based clustering Algorithm*

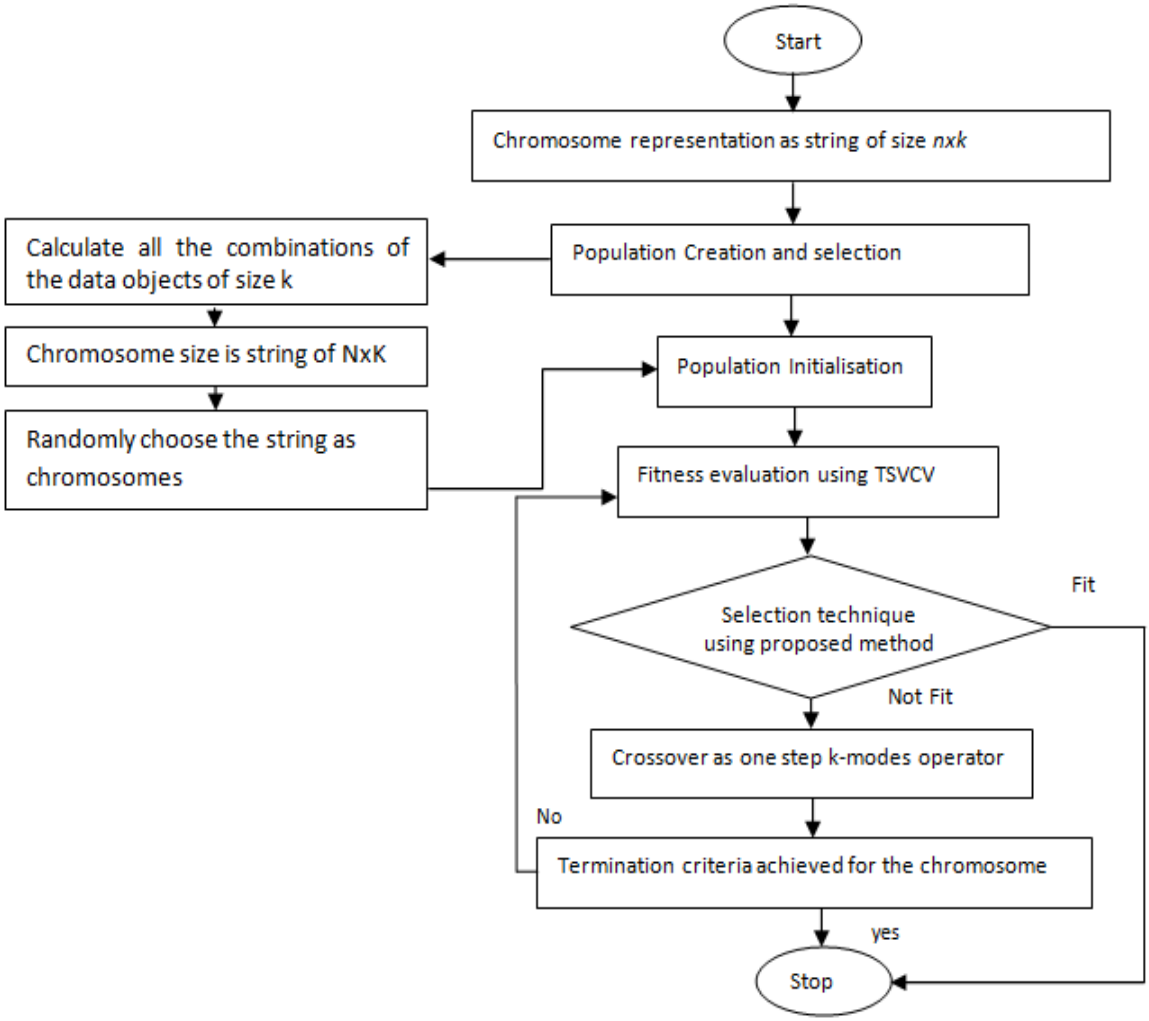

Fig. 2. Flow chart Proposed clustering Algorithm

V. EXPERIMENTAL RESULTS

In this work, the proposed GA based clustering algorithm and the standard *k*-Modes method were coded using python language. The experiments has been conducted on a computer laptop with 2.89 GHz CPU and 8 G RAM under a Windows 8.1 operating system. To test the effectiveness of the proposed algorithm on Soybean dataset from UCI [16] has been used.

Soybean dataset: The dataset contains 47 instances, 35 attributes, and 19 classes and four classes are considered in reality. And out of 35 attributes 14 attributes categories are same so we shall use 21 attributes only. Existing *k*-Modes algorithm has been run for 100 iterations with different initialisation say different seeds and different number of *k*. Proposed GA based clustering algorithm were executed 5 times for soybean dataset and *k*-Modes executed approx. 10-10 times for each *k* of the dataset. Proposed algorithm has been run till 100 iterations. To evaluate performance measure computational time (in seconds) has been calculated for algorithm efficiency.

Secondly, the TWCV is an intrinsic validity measure to calculate the sum of within cluster variation for all clusters. The smaller value of TWCV means the dataset are more compact. Therefore, in order to obtain compact clusters or mor separated clusters the value should be minimized for clustering task. If only considering the computational efficiency, the faster algorithm is better. The detailed analysis will be shown in the next sub-sections.

The shaded values shown in tables II-VII are locally optimal and globally optimal in case of *k*-Modes and proposed algorithm respectively.

The detailed clustering results of *k*-Modes algorithm for soybean data on different initialization with different *k* values has been shown in Table II-IV which shows the values are stuck at locally optimal. The proposed clustering algorithm provides the optimal values from table V-VII in all the runs for all the *k*. *K*-Modes algorithm also attains somewhere the optimal value as proposed value of the total runs but the ratio are very less. Table VIII, X, XII, XIV, XVI shows the average cost of different initialisation in 100^{th} iteration for different *k* using *k*-Modes. Table IX, XI, XIII, XV, XVII shows the average cost of different population in 100^{th} iteration for different *k* using proposed algorithm.

Fig. 3 shows the cost gap increases between the *k*-Modes and proposed method which shows the compact clustering of proposed algorithm. Fig. 4. shows the time obtained to cluster *k*-Modes and proposed method show the very less time gap between the proposed algorithm and *k*-Modes.

TABLE II. TWCV USING *K*-MODES ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS CLUSTER CENTRES WHEN K=2

Initial Configuration	i=1	i=2	i=3	i=4	i=5	i=6	i=100
1	23.89	19.29	16.37	16.34	16.34	16.34	16.34
2	20.96	19.29	18.25	17.7	16.93	16.24	16.24
3	20.99	17.18	17.07	17.07	17.07	17.07	17.07
4	22.22	16.78	16.35	16.34	16.34	16.34	16.34
5	21.42	19.81	18.19	17.41	17.07	16.34	16.34
6	25.3	19.35	19.22	18.89	17.89	17.87	17.87
7	29.13	17.88	17.07	16.34	16.34	16.34	16.34
8	24.28	17.33	16.5	16.5	16.5	16.5	16.5
9	25.033	17.93	17.07	16.34	16.34	16.34	16.34
10	31.53	17.93	17.07	16.34	16.34	16.34	16.34
11	20.29	19.22	19.22	19.22	19.22	19.22	19.22
12	18.23	17.15	17.15	17.15	17.15	17.15	17.15
13	7.35	17.07	17.07	17.07	17.07	17.07	17.07

TABLE III. TWCV USING *K*-MODES ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS CLUSTER CENTRES WHEN K=3

Initial Configuration	i=1	i=2	i=3	i=4	i=5	i=6	i=100
1	20.33	15.72	15.4	15.4	15.4	15.4	15.4
2	20.58	20.32	19.47	19.47	19.47	19.47	19.47
3	17.77	16.41	15.87	17.48	18.41	16.24	16.24
4	22.16	15.15	14.93	14.93	14.93	14.93	14.93
5	23.56	19.89	18.19	17.41	17.07	16.34	16.34
6	27.72	14.32	14.32	15.26	15.22	15.22	15.22
7	24.35	17.42	18.68	16.35	14.78	15.22	15.22
8	25.28	17.7	17.7	15.64	15.54	15.48	15.48
9	20.63	17.76	15.07	14.41	15.16	14.93	14.93
10	28.69	15.73	15.44	15.44	15.44	15.44	15.44
11	20.87	17.34	16.45	16.42	16.42	16.42	16.42
12	18.04	15.34	16.42	16.42	16.42	16.42	16.42

TABLE IV. TWCV USING K-MODES ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS CLUSTER CENTRES WHEN K=4

Initial Configuration	i=1	i=2	i=3	i=4	i=5	1=6	i=100
1	19.58	19.16	17.24	15.04	19.79	18.99	18.99
2	20.14	18.1	17.22	16.24	16.24	16.24	16.24
3	17.58	15.44	15.44	15.44	15.44	15.44	15.44
4	21.44	19.89	18.1	17.41	17.04	16.34	16.34
5	23.59	15.92	15.66	16.32	16.22	16.22	16.22
6	28	17	15.29	15.29	15.29	15.29	15.29
7	19.13	12.57	13.87	13.83	13.83	13.83	13.83
8	26.34	19.37	17.94	17.07	16.34	16.34	16.34
9	20.59	12.75	12.75	12.75	12.75	12.75	12.75
10	28.46	16.6	15.43	15.41	15.41	15.41	15.41
11	21.39	17.89	17.49	17.31	17.31	17.31	17.31
12	16.59	16.06	15.31	15.97	15.18	14.93	14.93
13	18.97	15.36	16.42	16.42	16.42	16.42	16.42

TABLE V. TWCV USING PROPOSED ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS POPULATION SIZE WHEN K=2

Initial Population	i=1	i=2	*i=3*	i=4	i=100
5	16.7	15.44	15.44	15.44	15.44
10	17.25	15.59	15.59	15.59	15.59
15	17.09	15.44	15.44	15.44	15.44
20	17.28	15.44	15.44	15.44	15.44
100	16.92	15.44	15.44	15.44	15.44

TABLE VI. TWCV USING PROPOSED ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS POPULATION SIZE WHEN K=3

Initial Population	i=1	i=2	i=3	i=4	i=100
5	11.99	11.06	11.06	11.06	11.06
10	15.16	11.06	11.06	11.06	11.06
15	16.88	11.29	11.06	11.06	11.06
20	15.05	11.06	11.06	11.06	11.06
100	11.58	11.14	11.06	11.06	11.06

TABLE VII. TWCV USING PROPOSED ALGORITHM ON VARIOUS ITERATIONS WITH VARIOUS POPULATION SIZE WHEN K=4

Initial population	i=1	i=2	i=3	i=4	i=100
5	15.23	10.97	10.55	10.55	10.55
10	12.29	10.55	10.55	10.55	10.55
15	13.37	8.60	8.60	8.60	8.60
20	11.78	11.15	11.14	11.14	11.14
100	11.40	10.28	10.47	10.47	10.47

TABLE VIII. AVERAGE TWCV OBTAINED USING K-MODES ALGORITHM FOR DIFFERENT INITIALISATION AFTER 100 ITERATION WHEN *K*=2

Initial configuration	*k*-Modes
1.	18.10
2.	19.22
3.	17.07
4.	17.07
5.	16.34

TABLE IX. AVERAGE TWCV OBTAINED USING PROPOSED ALGORITHM FOR DIFFERENT POPULATION SIZE AFTER 100 ITERATION WHEN *K*=2

Initial population	Proposed algorithm
5	15.44
10	15.44
15	15.44
20	15.44
100	15.44

TABLE X. AVERAGE TWCV OBTAINED USING K-MODES ALGORITHM FOR DIFFERENT INITIALISATION AFTER 100 ITERATION WHEN K=3

Initial configuration	*k*-Modes
1.	15.44
2.	19.47
3.	15.48
4.	16.24
5.	16.24

TABLE XI. AVERAGE TWCV OBTAINED USING PROPOSED ALGORITHM FOR DIFFERENT POPULATION SIZE AFTER 100 ITERATION WHEN *K*=3

Initial population	Proposed algorithm
5	11.06
10	11.06
15	11.06
20	11.06
100	11.06

TABLE XII. AVERAGE TWCV OBTAINED USING K-MODES ALGORITHM FOR DIFFERENT INITIALISATION AFTER 100 ITERATION WHEN K=4

Initial configuration	*k*-Modes
1.	14.80
2.	16.18
3.	10.62
4.	15.22
5.	15.22

TABLE XIII. AVERAGE TWCV OBTAINED USING PROPOSED ALGORITHM FOR DIFFERENT POPULATION SIZE AFTER 100 ITERATION WHEN *K*=4

Initial population	Proposed algorithm
5	10.55
10	10.55
15	8.60
20	11.14
100	10.47

TABLE XIV. AVERAGE TWCV OBTAINED USING K-MODES ALGORITHM FOR DIFFERENT INITIALISATION AFTER 100 ITERATION WHEN K=5

Initial configuration	*k*-Modes
1.	18.99
2.	16.24
3.	17.31
4.	16.34
5.	16.34

TABLE XV. AVERAGE TWCV OBTAINED USING PROPOSED ALGORITHM FOR DIFFERENT POPULATION SIZE AFTER 100 ITERATION WHEN *K*=5

Initial population	Proposed algorithm
5	11.35
10	9.70
15	11.14
20	10.39
100	9.24

TABLE XVI. AVERAGE TWCV OBTAINED USING K-MODES ALGORITHM FOR DIFFERENT INITIALISATION AFTER 100 ITERATION WHEN K=6

Initial configuration	*k*-Modes
1.	14.12
2.	12.89
3.	16.22
4.	18.99
5.	16.22

TABLE XVII. AVERAGE TWCV OBTAINED USING PROPOSED ALGORITHM FOR DIFFERENT POPULATION SIZE AFTER 100 ITERATION WHEN *K*=6

Initial population	Proposed algorithm
5	10.87
10	12.46
15	9.45
20	11.93
100	9.05

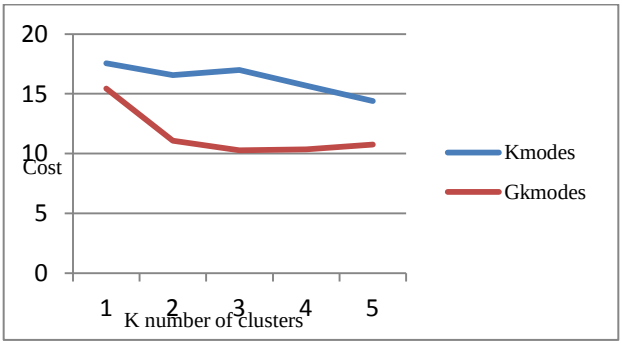

Fig. 3. Comparison of average cost obtained by proposed algorithm and k-modes algorithm for different k after 100 iteration

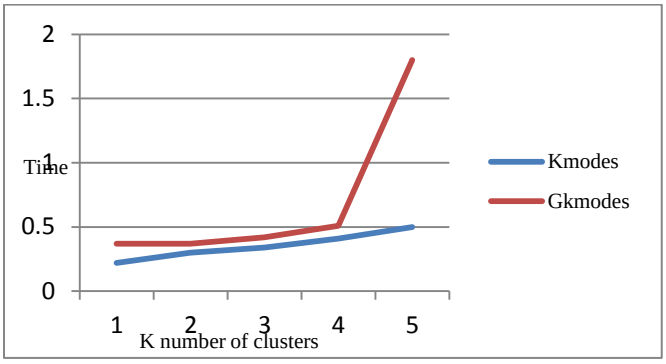

Fig. 4. Comparison of total time obtained by proposed algorithm and k-modes algorithm for different k after 100 iteration

VI. CONCLUSION

Many clustering results are sensitive to the selection of the initial cluster centres as well as gives local optimal solution. The determination of cluster centres in a data set is attracting attention in many research areas. This paper introduced a new variant of GA based clustering for categorical data with the analysis of local and global optimality with *k*-Modes. Existing approaches does not serve as the best method in terms of time and space, Experiments proves noticeably results in terms of cost, within cluster variation, time and initialization of cluster centres.

VII. FUTURE WORK

As this work gives better results for less number of clusters using MATLAB [17]. This can be modify if the number of clusters increased. Proposed method can be compared to more recent algorithms with more number of real world datasets. To discover an algorithm which can perform clustering without knowing cluster number is also a significant work in clustering analysis can be done. And to increase the convergence speed is an important area of future research. Using GA on large number of attributes in datasets need more time and space so latest feature selection techniques [18] can also be applied.

REFERENCES

[1] Han, J. ; Kamber, M. (2001): Data Mining: Concepts and Techniques, Morgan Kaufmann Publisher, San Francisco, CA.

[2] Dongxia Chang, Yao Zhao , Changwen Zheng, Xianda Zhang, A genetic clustering algorithm using a message-based similarity measure Expert Systems with Applications, 39, (2012), 2194–2202.

[3] Sneha Antony, Jayarajan J N , T-GEN: A Tabu Search based Genetic Algorithm for the Automatic Playlist Generation Problem, Procedia Computer Science 46 (2015) 409 – 416

[4] Z. Huang, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets inData Mining", 1997.

[5] B. Andreopoulos, A. An, X. Wang, "Hierarchical Density-Based Clustering of Categorical Data and a Simplification", PAKDD, 2007, pp. 11–22.

[6] T. Chen, N.L. Zhang, Y. Wang, "Efficient model evaluation in the search-based approach to latent structure discovery", Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM-08), Vol. 8, 2008, pp. 57–64.

[7] Yinghua Lv, Tinghuai Ma , Meili Tang, Jie Cao, Yuan Tian , Abdullah Al-Dhelaan, Mznah Al-Rodhaan, An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22.

[8] Arkajyoti Saha, Swagatam Das, Categorical fuzzy *k*-Modes clustering with automated feature weight learning, Neurocomputing 166 (2015) 422–435.

[9] E. David, Goldberg, H. Holland John, "Genetic Algorithms and Machine Learning, Machine Learning", Vol. 3,1988. pp. 95-99.

[10] C. L. Yang, R. J. Kuo, C. H. Chien, N. T. P. Quyen, "Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering", Applied Soft Computing, Vol. 30, 2015, pp 113–122.

[11] H. Qin, X. Ma, T. Herawan , J.M. Zain, "An Improved Genetic Clustering Algorithmfor Categorical Data", In PAKDD Workshops, LNAI, Vol. 7769, 2013, pp. 100–111.

[12] Y. Liu, X. , Y.Shen, "Automatic clustering using genetic algorithms", Applied Mathematics and Computation, Vol. 218, 2011, pp. 1267–1279.

[13] S. Deng, Z. He, X. Xu, "G-ANMI: A mutual information based genetic clustering algorithm for categorical data", Knowledge-Based Systems, Vol. 23, 2010, pp. 144-149.

[14] U. Maulik, S. Bandyopadhyay, "Genetic algorithm-based clustering technique", Pattern Recognition, Vol. 33, 2001, pp. 1455-1465.

[15] G. Gan, Z. Yang, J. Wu, "A Genetic *k*-Modes Algorithm for Clustering", In ADMA, LNAI Vol. 3584, 2005, pp. 195–202.

[16] UCI Machine Learning Repository (2011). http://www.ics.uci.edu_/mlearn/MLRepository.html

[17] Abha Sharma, R. S. Thakur, Cluster analysis for categorical data using MATLAB, International Journal of Research in Management, Science & Technology 2014, Vol2, No. 2, pp 65-68.

[18] Hari Seetha; M. Narasimha Murty; R. Saravanan, Effective feature selection technique for text classification, Int. J. of Data Mining, Modelling and Management , 2015, Vol.7, No.3 , pp.165 - 184.