Hybridized Improved Genetic Algorithm with Variable Length Chromosome for Image Clustering

Venkatesh Katari, *Suresh Chandra Satapathy, *Member IEEE* ,**
****JVR Murthy, ***PVGD Prasad Reddy**
GITAM Engineering College, *ANITS College of Engineering, Visakhapatnam,
** JNTU College of Engineering, Kakinada, *** AU Engineering College, Visakhapatnam

Abstract

Clustering is a process of putting similar data into groups. This paper presents data clustering using improved genetic algorithm (IGA) in which an efficient method of crossover and mutation are implemented. Further it is hybridized with the popular Nelder-Mead (NM) Simplex search and K-means to exploit the potentiality of both in the hybridized algorithm. The performance of hybrid approach is evaluated with few data clustering problems. Further a Variable Length IGA is proposed which optimally finds the clusters of benchmark image datasets and the performance is compared with K-means and GCUK[12].The results revealed are very encouraging with IGA and its hybridization with other algorithms.

Key words:
K-means, Nelder-Mead, Variable length genetic algorithm

1. Introduction

Data clustering is an important Data Mining task and several clustering algorithms are proposed in [1][2]. Among them, the K-means [3] algorithm is an important one. It is an iterative hill climbing algorithm and the solution obtained depends on the initial random selection of cluster centroids. Although the K-means algorithm had been applied to many practical clustering problems successfully, it has been shown that the algorithm may fail to converge to a global minimum under certain conditions. Since the genetic algorithm [4] is good at searching, this can be applied to search the optimal cluster. The GA has been proposed to find suitable clusters in [5][6]. One of the demerits of GA based clustering is that it takes large computation time in converging to optimal solution. In this paper we implement an improved genetic algorithm (IGA) for data clustering which involves far less computations then the standard genetic algorithm [4].The improved GA performs more efficiently compared to standard GA. The convergence rate of GA is also typically seen to be slower than those of local search techniques such as Nelder-Mead(NM) Simplex Search [9]. To deal with the slow convergence two hybrid algorithms based on the combination of Nelder-Mead search and Improved GA and K-means, Nelder-Mead and IGA are proposed and experimentation done on them for clustering on few data sets for which the number of clusters are known beforehand. The major challenge in image clustering is to find the optimal cluster numbers of the image. In[12] authors have proposed a variable GA known as GCUK(Genetic Clustering with Unknown K values) where K is the number of clusters to find the optimal clusters of images data sets. In this paper we have implemented a novel variable length IGA (VLIGA) for image clustering which automatically finds the optimal numbers of cluster and the results are compared with K-means and GCUK [12] with a modified mutation function. The results are found to be very competitive and efficient when compared with K-means and GCUK with modified mutation.

The rest of the paper is organized as follows. In section 2 the improved genetic algorithm is presented. In section 3, two hybrid designs are proposed combining NM and IGA and NM, IGA and K-means. In section 4 the frame work for data clustering using the proposed algorithms are discussed. Experimental results are given in Section 5. Section 6 explains variable length genetic algorithm. Simulation study on image clustering is presented in section 7. Section 8 gives conclusion and future research.

2. Improved GA

Genetic algorithms (GA) [4] are an evolutionary optimization approach which is an alternative to traditional optimization methods. GA is most appropriate for complex non-linear models where location of the global optimum is a difficult task. The different genetic operators and the standard GA approach is described in [4]. The traditional GA as described in [4] and implemented for clustering problem in [5][12][13] found to take more time to converge to good acceptable solution. In this paper we have implemented a modified crossover and mutation technique [8] for data clustering problems which make the convergence faster compared to GA described in [4]. Our modified GA is known as IGA here. The modified

Manuscript received November 5, 2007
Manuscript revised November 20, 2007

crossover and mutation techniques of our IGA are described below.

A. Crossover:

The initial population is a potential solution set **P.** The first set of population is usually generated randomly and is denoted by

$$P=\{ P_1 , P_2 , P_3, \ldots\ldots\ldots\ldots, P_{pop\text{-}size} \}$$

$$\text{Where } P_i =[p_{i1}, p_{i2}, p_{i3} \ldots\ldots\ldots p_{ij}\ldots\ldots p_{ino\_vars}]$$

$$i= 1,2,\text{-----}, pop\_size$$
$$j= 1,2, \text{------}, no\_vars$$

pop_size and no_vars denote the population size and number of variables to be tuned respectively. These two parents are used to produce four offspring as given below.

$$os^1{}_c = [os^1{}_1 \ os^1{}_2 \ldots\ldots os^1{}_{no\_vars}]$$
$$= (P1, P2) / 2 \quad \text{---------- (1)}$$

$$os^2{}_c = [os^2{}_1 \ os^2{}_2 \ldots\ldots os^2{}_{no\_vars}]$$
$$= Pmax(1-w) + max(P1,P2)w \quad \text{---------- (2)}$$

$$os^3{}_c = [os^3{}_1 \ os^3{}_2 \ldots\ldots os^3{}_{no\_vars}]$$
$$= Pmin(1-w) + min(P1,P2)w \quad \text{----------(3)}$$

$$os^4{}_c = [os^4{}_1 \ os^4{}_2 \ldots\ldots os^4{}_{no\_vars}]$$
$$= ((Pmax + Pmin) (1-w) + (P1,P2)w) / 2 \quad \text{--------- (4)}$$

where

$$Pmax = [para^1{}_{max} \quad para^2{}_{max} \ldots\ldots\ldots para^{no\_vars}{}_{max}]$$

$$Pmin = [para^1{}_{min} \quad para^2{}_{min} \ldots\ldots\ldots para^{no\_vars}{}_{min}]$$

and $para^j{}_{max}$ and $para^j{}_{min}$ are the maximum and minimum values of the parameters p_i respectively for all i . The w denotes the weight which can be determined by users in the range 0 to 1. These four offspring are evaluated as per the fitness function of the problem and one with the largest fitness value is used as the offspring of the crossover operation in our work. This offspring can be defined as

$$os=[os_1 \quad os_2 \quad \ldots\ldots\ldots os_{no\_vars}] \quad \text{---------- (5)}$$

B. Mutation:

The offspring chosen by the above crossover approach will then undergo mutation. In this case equation (5) undergoes mutation operation. In our implementation three new offspring are generated by mutation operation. It is denoted by

$$nos_j = [os_1 \quad os_2 \quad \ldots\ldots\ldots os_{no\_vars}] +$$
$$[b_1 \Delta.nos_1 \ b_2 \Delta.nos_2 \ldots b_{no\_vars} \Delta.nos_{no\_vars}]$$
$$j=1,2,3 \quad \text{---------- (6)}$$

where bi, i = 1, 2, ….., no_vars, can only take the value of 0 or 1 , $\Delta.nos_i$, i = 1,2, ..., no_vars are randomly generated numbers such that $para^i{}_{min} <= os_{i\ +}\ \Delta.nos_i <= para^i{}_{max}$. Three new offspring are generated as detailed below.

The first new offspring (j = 1) is obtained according to (6) with that only one bi (i being randomly generated within the range) is allowed to be one and all the others are zeros. The second new offspring is obtained according to (6) with that some bi randomly chosen are set to be one and others are zero. The third new offspring is obtained according to (6) with all bi = 1.

These three new offspring will then be evaluated using the fitness function of as per the given problem and will be appended to the main population and the main population is sorted with regard to fitness values and the first population number of chromosomes are selected as parents for next generation. A real number will be generated randomly and compared with a user-defined number Pa which is probability of acceptance and which is to be chosen between 0 and 1. If the real number is smaller than Pa, then a chromosome is randomly selected from the discarded population and is made to replace the last chromosome in the population, this is done because a worst parent at that generation may give a good child in future generation. Pa is effectively the probability of accepting a bad offspring in order to reduce the chance of converging to a local optimum. Hence, the possibility of reaching the global optimum is kept.

3. Hybrid Algorithms

In this section we present implementation strategies of hybridization approaches of K-menas, Nelder –Mead and IGA. K-means reported to be faster but the clustering result is poor as it depends on initial seed values. Nelder-Mead on the other hand has better convergence rate compared to GA. In this study we have carefully explored the merits of each technique to propose hybridized approaches. The implementation strategies of these approaches for data clustering problems are briefly explained below.

3.1Hybrid NM–IGA:

The population size of this hybrid *NM–IGA* approach is set at 3N + 1 when solving an N-dimensional problem. The initial 3N + 1 particles are randomly generated and sorted by fitness, and the top N + 1 particles are then fed into the simplex search method to improve the (N + 1) th particle [9]. The other 2N particles are adjusted by the *IGA* method

using the modified crossover and mutation mechanisms explained above. The 3N + 1 particles are sorted again in preparation for repeating the entire run. The process terminates when certain convergence criteria are met.

3.2 Hybrid KM–NM–IGA:

The K-means algorithm tends to converge faster than the IGA as it requires fewer function evaluations, but it usually results in less accurate clustering. One can take advantage of its speed at the inception of the clustering process and leave accuracy to be achieved by other methods at a later stage of the process. This statement shall be verified in later sections of this paper by showing that the results of clustering by IGA can further be improved by seeding the initial population with the outcome of the K-means algorithm (denoted as *KM–IGA* and *KM–NM–IGA*). More specifically, the hybrid algorithm first executes the K-means algorithm, which terminates when there is no change in centroid vectors. In the case of *KM–IGA*, the result of the K-means algorithm is used as one of the chromosomes, while the remaining chromosomes are initialized randomly. The *IGA* algorithm then proceeds as presented above. In the case of *KM–NM–IGA*, the first chromosome is seeded from k-means algorithm and rest 3N particles or vertices as termed in[9]*)* are randomly generated and *NM–IGA* is then carried out to its completion.

4. Framework for Data clustering using improved GA

The following symbols are defined for the purpose of explaining our paper

N_d: the input dimension,
No: the number of data vectors to be clustered
N_c: the number of cluster centroids i.e. the number of clusters to be formed
z_p: the p^{th} data vector
M_j: the centroid vector of cluster j
nj: the number of data vectors in cluster j
Cj: the subset of data vectors that form cluster j.
Z_{max} : Value of the maximum data element in the dataset
d_{intra} : Intra cluster distance
D_{inter} : Inter cluster distance
w1,w2,w3: Weights

The improved GA maintains a population of chromosomes, where each chromosome represents a potential solution to an optimization problem. In the context of clustering, a single chromosome represents the N_c cluster centroid vectors. That is, each chromosome x_i is constructed as follows:

$$x_i = \left(m_{i1}, m_{i2} \ldots\ldots\ldots m_{ij} \ldots\ldots\ldots m_{iN_c}\right) \text{------- (7)}$$

where m_{ij} is the j^{th} cluster centroid vector of the i^{th} chromosome in cluster C_{ij}. The fitness of each chromosome is measured by the two different approaches, one by calculating the quantization error and other by the computation of intra and inter- cluster distances. The quantization error is given by

$$Q_e = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall z_p \in c_{ij}} \frac{d\left(z_p, m_j\right)}{\text{mod}(c_{ij})}\right]}{N_c} \text{---------- (8)}$$

where *d* is defined as the Euclidian distance between each data vector and the centroid of the cluster and is given by

$$d\left(z_p, m_j\right) = \sqrt{\sum_{k=1}^{N_d}\left(z_{pk} - m_{jk}\right)^2} \text{---------- (9)}$$

$\text{mod}(C_{ij})$ is the number of data vectors belonging to cluster C_{ij} i.e., the frequency of that cluster. The other fitness function is given as

$$fit = \min(w_1 * d_{\text{int } ra} + w_2 * (N_c * Z_{\max} - d_{\text{int } er}) + w_3 * Q_e$$

---------- (10)

The objective is to improve the compactness of each cluster by minimizing the intra-cluster distances and improving the separation among clusters by maximizing the inter-cluster distance along with minimizing the quantization error.

5. Simulation results

The clustering problems used for the purpose of this paper are collected from UCI machine repository:

Artificial data set one: (N_o = 250, N_d = 3, N_c = 5): This is a three-featured problem with five classes, where every feature of the classes was distributed according to Class1~uniform(85,100),Class2~Uniform(70,85),Class3~ Uniform(55,70), Class4~ Uniform(40, 55), Class5~ Uniform(25, 40). The data set is illustrated in Fig. 2

***Artificial dataset two*:** (N_o = 400, N_d = 2, N_c =2)This problem follows the following classification rule:

$$\text{class} = \begin{cases} 1 & \text{if } (z_1 \geq 0.7) \text{ or } ((z_1 \leq 0.3) \\ & \text{and } (z_2 \geq -0.2 - z_1)) \\ 0 & \text{otherwise} \end{cases}$$

A total of 400 data vectors were randomly created, with $z_1, z_2 \in (-1,1)$. This dataset is illustrated in fig. 3

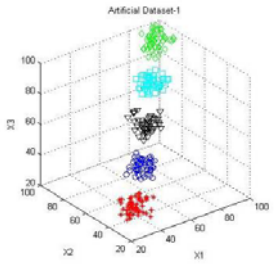

Fig.2. Artificial Dataset 1

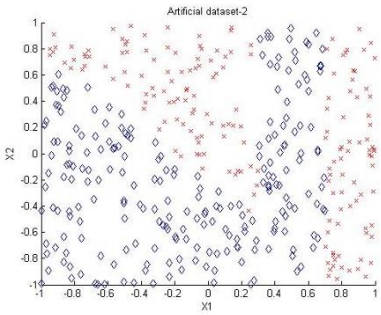

Fig.3. Artificial Dataset 2

5.1. Results

The main purpose of this section is to compare the quality of the respective clustering based on the quantization error (8), and fitness function (10).The Nelder-Mead Simplex Search algorithm is efficient and fast but it has a high probability of getting trapped in Local minima like the traditional K-means algorithm. But it can perform better when it has a good initial particle. When compared to IGA, the Nelder-Mead algorithm performs better in the case of HYES dataset. For all the remaining datasets, IGA is a good performer. The performance of IGA clustering can be further enhanced by seeding the initial population by the result of K-means algorithm. Since NM is a fast local search technique, the performance of KM-IGA can be improved very much by hybridizing KM-IGA and NM. For our simulation the w and p_a (for IGA) values are set to 0.5 and 0.1 respectively, the values taken for w1, w2 and w3 in equation (10) are 0.5, 0.5 and 0.3 respectively. Each algorithm is run for 30 simulations and 200 iterations. The values are reported for two fitness measures with standard deviations. Considering the quantization error first, it can be verified from the table 3 that in some cases all 3 algorithms IGA, KM-IGA, KM-NM-IGA have same Qe but the KM-NM-IGA hybrid algorithm has the least standard deviation. However, if the other fitness function i.e. *fit* is considered hybrid algorithm KM-NM-IGA, KM-IGA and IGA gave equal fitness for Artificial Dataset 2 and again the KM-NM-IGA hybrid has least standard deviation. The *fit* values given by K-Means for IRIS and WBC are relatively close to the fitness values given by the rest of the algorithms , in all other cases the IGA, KM-IGA and KM-NM-IGA are far more better. It can also be inferred from the graphs below that the performance of the 3 algorithms are much better compared to K-means. Figure 9 shows that these 3 algorithms perform equally for the artificial dataset 2. Figure 5 shows that both KM-IGA and KM-NM-IGA performs almost equally for artificial data set 1.The results show the general improvement of hybrid KM-NM-IGA when it is seeded with the outcome of the K-means.

Data Set	Criteria	K-means	Nelder- Mead (NM)	IGA	KM-IGA	K-NM-IGA
Art1	Fitness Best	1715.6259	1239.2986	949.7263	909.1329	907.4211
	Std(Fitness)	156.9837	31.4932	39.9159	8.4840	4.1431
	Qe	7.8194	9.2141	7.0519	7.0479	7.0479
	Std(Qe)	1.5683	0.7988	0.2614	0.1629	0.000
Art2	Fitness Best	125.9777	122.9351	122.8913	122.8913	122.8913
	Std(Fitness)	10.7832	0.4904	0.2011	0.1490	0.0216
	Qe	0.6022	0.6044	0.6045	0.6045	0.6039
	Std (Qe)	0.032	0.0039	0.0011	0.0013	0.00027
IRIS	Fitness Best	55.4897	53.2701	53.1352	53.129	52.9211
	Std(Fitness)	1.9501e-04	0.8241	0.0219	0.0173	0.0109
	Qe	0.6525	0.6494	0.6463	0.6404	0.6343
	Std(Qe)	1.498e-06	0.2436	0.1423	0.1365	0.0110

WINE	Fitness Best	10337.2215	9241.6427	9227.6453	9224.507	9222.4211
	Std(Fitness)	0.0763	18.7763	41.7309	32.6175	9.6305
	Qe	107.223	97.739	97.517	97.517	97.517
	Std(Qe)	0.0010	0.5643	0.1623	0.0658	0.0232
BREAST CANCER	Fitness Best	1545.038	1543.6627	1532.5397	1531.9396	1531.009
	Std(Fitness)	4.796e-8	12.5631	1.8794	0.4294	0.4001
	Qe	5.2526	5.2199	5.1799	5.1799	5.1913
	Std(Qe)	9.3622e-12	0.5972	0.0653	0.0454	0.0277
HAYES	Fitness Best	885.5343	823.4388	823.6242	822.5936	821.291
	Std(Fitness)	2.973e-11	3.8009	2.5247	1.1905	0.6478
	Qe	11.296	11.2651	11.2627	11.2627	11.2627
	Std(Qe)	1.324e-15	0.8326	0.1264	0.08	0.0041
DIABETES	Fitness Best	26036.6128	25153.922	25145.62	25136.86	24976.0536
	Std(Fitness)	7.69e-12	246.4682	17.5198	12.2521	6.3725
	Qe	77.1106	70.2655	70.301	70.2019	70.2006
	Std (Qe)	0.0001	3.5498	0.4657	0.2921	0.1669

Table 3: Simulation results after 30 runs

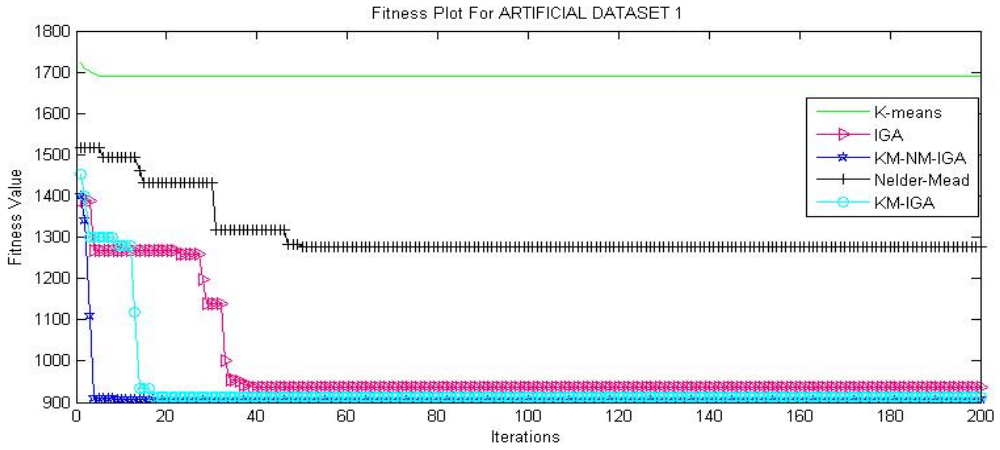

Fig.4. Convergence Plot of Fitness for Artificial Dataset 1

5.2. Cluster plots for Artificial Dataset 1:

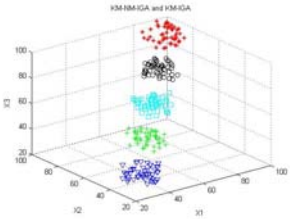

Fig.5. KM-NM-IGA, KM-IGA

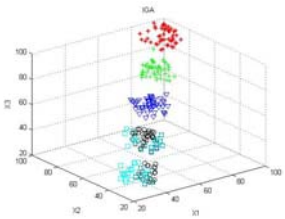

Fig.6. IGA

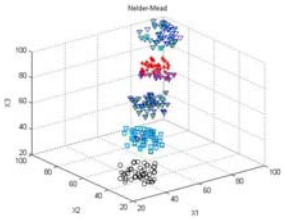

Fig.7. Nelder-Mead

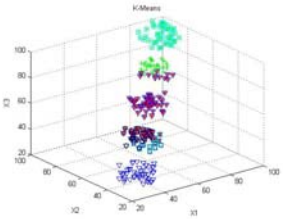

Fig.8. K-Means

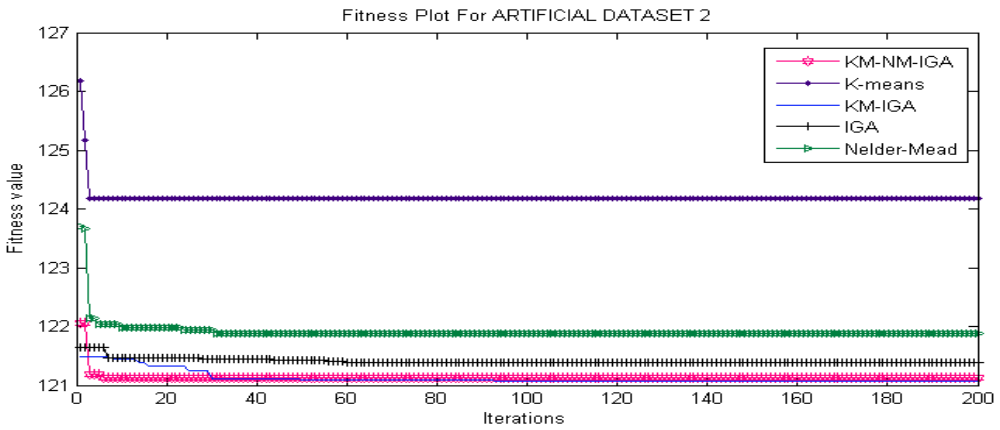

Fig.9. Convergence Plot of Fitness for Artificial Dataset 2

5.3. Cluster Plot for Artificial dataset 2

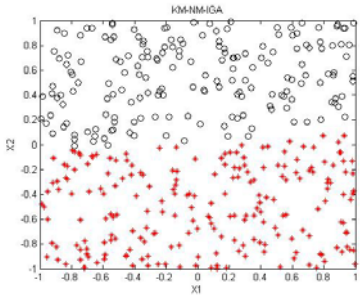

Fig.10.KM-NM-IGA

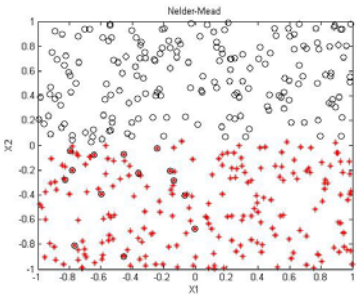

Fig.11. NM

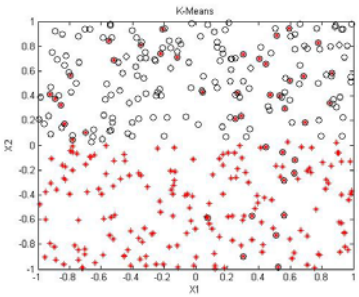

Fig.12: K-MEANS

6. Variable Length IGA (VLIGA) for Image Clustering

One can easily infer from section 5 that IGA itself can also be used for clustering purposes and its results are satisfactory. Now in this section we try to apply evolutionary partitional clustering algorithms Variable length IGA (VLIGA) and GCUK [12] to the segmentation of gray scale images, the intensity level of each pixel serve as a feature for the clustering process. The following sections describe the genetic operators implemented in our study.

6.1Chromosome representation

In GA applications, the unknown parameters are encoded in the form of strings, so-called chromosomes. A chromosome is encoded with binary, integer or real numbers. Since multispectral image data are usually represented by positive integers, in this research a chromosome is encoded with a unit (tuple) of positive integer numbers. Each unit represents a combination of brightness values, one for each band, and thus a potential cluster centroid. The length of the chromosome, *K*, is equivalent to the number of clusters in the classification problem. *K* is selected from the range [*Kmin*, *Kmax*], where *Kmin* is usually assigned to 2 unless special cases are considered [12] , and *Kmax* describes the maximum chromosome length, which means the maximum number of possible cluster centroids. *Kmax* must be selected according to experience. Without assigning the number of clusters in advance, a variable string length is used. Invalid (non-existing) clusters are represented with negative integer "NaN(not a number)". The values of the chromosomes are changed in an iterative process to determine the correct number of clusters (the number of valid units in the chromosomes) and the actual cluster centroids for a given classification problem.

6.2 Chromosome initialization

A population is the set of chromosomes. The typical size of the population can range from 20 to 1000. In the following an example is given to explain the creation of an initial population: we assume to have a satellite image with three bands, *Kmin* is set to 2 and *Kmax* to 8. At the beginning, for each chromosome *i* (*i* =1, 2,…,.*P*, where *P* is the size of population) all values are chosen randomly from the data space (universal data set; here: positive integers). Such a chromosome belongs to the so-called parent generation. One (arbitrary) chromosomes of the parent generation is given here:

Nan (55) (150,246,23) Nan (11) Nan Nan (100)

6.3 Crossover and Mutation

Crossover: The purpose of the crossover operation is to create two new individual chromosomes from two existing chromosomes selected randomly from the current population. Typical crossover operations are one-point crossover, two-point crossover, cycle crossover and uniform crossover. In this research, only the simplest one, the one-point crossover was adopted; the following example illustrates this operation (the point for crossover is after the 4th position):
Parent1 : Nan (88) (226) Nan (104) (50) Nan (192)
Parent2 : (127) (88) Nan Nan (45) Nan (174) (101)
Child1 : Nan (88) (226) Nan (45) Nan (174) (101)
Child2 : (127) (88) Nan Nan (104) (50) Nan (192)

Mutation
The non-uniform mutation operator is applied in this study in contrast to the mutation operation described in [12]. It selects one of the parent chromosome genes gi and adds to it a random displacement. The operator uses two uniform random numbers r1 and r2 drawn from the interval [0,1]. The first (r1) is used to determine the direction of the displacement while the other (r2) is used to generate the magnitude of the displacement. Assuming that $g_i \in [a_i, b_i]$, where a_i and b_i are the gene lower and upper bounds, respectively, the new variable becomes

$$q_i = \begin{cases} g_i + (b_i - g_i) f(G), r_1 < 0.5 \\ g_i - (g_i - a_i) f(G), otherwise \end{cases} \text{----------(11)}$$

where $f(G) = [r_2(1 - (G/G_{max}))]^p$, G is the current generation, G_{max} is the maximum number of generations, and p is a shape parameter.

6.4 Fitness Function (Davies Bouldin index)

Based on crossover and mutation the chromosomes, once initialized, iteratively evolve from one generation to the next. In order to be able to stop this iterative process, a fitness function needs to be defined to measure the fitness or adaptability of each chromosome in the population. The population then evolves over generations in the attempt to minimize the value of fitness, also called index. Previous research used different indices, such as distance, separation index, Fuzzy C-Means, Davies-Bouldin Index (DBI), and Xie-Beni Index (XBI)[16], as criteria to determine the best clustering[12]. Here, the DBI was adopted, because it is not as complex as fuzzy C-Means and one can obtain better results than with some other indices as shown using simulated data [13][15].
The fitness of a chromosome is computed using the Davies–Bouldin index [14]. This index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within Ci , the ith cluster, is computed as

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in c_i} \{ ||x - z_i||_2^q \} \right)^{1/q} \text{--------(12)}$$

where, zi is the centroid of Ci , and is defined as $Z_i = 1/n_i \sum_{x \in Ci} x$ and ni is the cardinality of Ci , i.e., the number of points in cluster Ci . The distance between cluster Ci and Cj is the Euclidean distance between them(9). Subsequently we compute

$$R_{i,qt} = \max_{j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\} \text{----------(13)}$$

The Davies-Bouldin(DB) index is then defined as

$$DB = 1/k \sum_{i=1}^{k} R_{i,qt} \text{----------(14)}$$

Now, the objective is to minimize the DB index for achieving proper clustering.

6.5 Variable Length IGA

We have used the same chromosome representation and crossover operation used in GCUK [12] but for doing mathematical operations between a integer centroid value and NaN we used the following logic . We generated a random number σ between [0 1] and if the value of σ is greater than 0.5 then we take the integer centroid as resultant gene in the child chromosome., and when the value of σ is less than 0.5 NaN is taken as the resultant gene in the child chromosome. The NaN or integral value in the child gene occurs with equal probability, hence the natural randomness of evolution is preserved.

7. Simulation Studies

In this section we report the results of applying 2 evolutionary partitional clustering algorithms such as VLIGA and GCUK and the traditional K-means algorithm to the segmentation of three 256×256 gray scale images. The intensity level of each pixel serves as a feature for the clustering process. For GCUK we have used crossover rate =80% and mutation rate=0.02%. Figures 13 to 24 show the three original images and their segmented counterparts obtained using the VLIGA, GCUK and the K-means based clustering algorithms. In table 4, we have reported the best classification results achieved with this database using the above algorithms.

IMAGES		VLIGA	GCU K	K-means
PEPPERS	DB	0.5192	0.5343	0.5498
	No. of clusters	7	4	7
LENA	DB	0.5203	0.5309	0.5498
	No. of clusters	5	5	5
CAMERA MAN	DB	0.4262	0.4623	0.6050
	No. of clusters	5	4	5

Table 4: Automatic clustering results using DBI

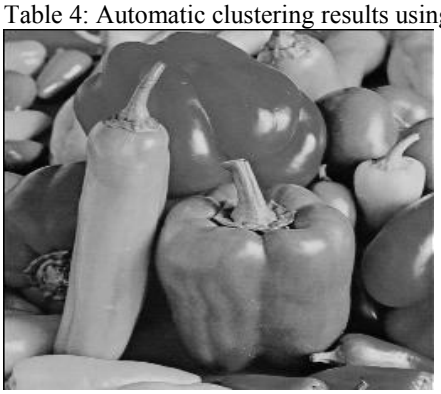

Fig .13.Original Peppers Image

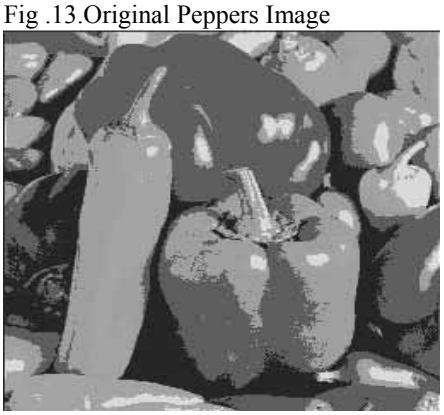

Fig.14. Clustering by VLIGA (k=7)

Fig.15. Clustering by GCUK (k=4)

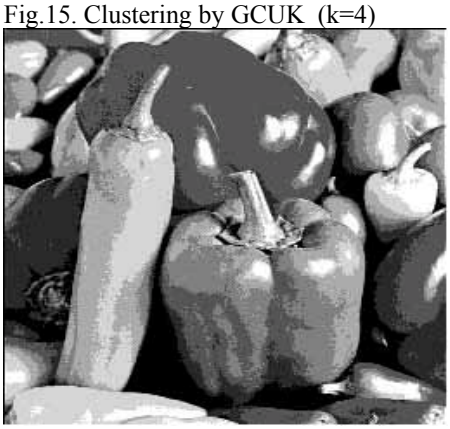

Fig .16. Clustering by k-means (provided k=7)

Fig.17. Original Lena Image

Fig.18. Clustering by VLIGA (k=5)

Fig.19. Clustering by GCUK (k=5)

Fig.20. Clustering by k-means(provided k=5)

Fig.21. Original Cameraman Image

Fig.22. Clustering by VLIGA (k=5)

Fig.23. Clustering by GCUK (k=4)

Fig.24. Clustering by k-means(provided k=5)

As one may see, GCUK algorithms fail to achieve optimal number of clusters for the peppers image which is verified from table IV. Figures 18-22 show original lena image and clustered images of VLIGA, GCUK & K-means respectively. From the results one can see that the background is not clearly separated out in case of GCUK, whereas with the VLIGA and k-means it is satisfactory. The shade on the face of Lena is well clustered with the proposed VLIGA algorithm which is not obtained with that of K-means. Also, the top left part of the background is well segmented by the VLIGA algorithm in comparison with the other two. From figures 21-24, it is possible to observe that GCUK performs better in terms of achieving homogeneity in grass area compared with the clustered image of VLIGA, but the GCUK algorithms fails to cluster the sky and background buildings properly and even the DB index achieved by it is low compared to VLIGA. K-means algorithm clustered out the sky and buildings but the amount of false classification was very high. It is evident from the figure 22 that the tripod and the background buildings are well clustered with the proposed VLIGA algorithm.

8. Conclusion and further enhancement

In this paper we have explored the capability of an improved GA based clustering on some well known data sets. Although K-means clustering is a very well established approach, however it has some demerits of initialization and falling in local minima. GA being a randomized based approach has the capability to alleviate the problems faced by K-means. In this paper an improved version of GA was discussed and implemented for data clustering. In this improved version of GA (IGA) a new approach of crossover and offspring formation adopted. When applied to data clustering problem IGA performs better compared to K-means in all data set under study in this paper. However, to further improvise the performance of IGA on data clustering the K-means was hybridized resulting in KM-IGA and boost the KM-IGA further more it has been hybridized with Nelder-Mead resulting in KM-NM-IGA. In hybrid algorithm (KM-NM-IGA) the outcome of K-means becomes one of the chromosomes in the initial population of NM-IGA. The results reveal that hybrid algorithm gives better results compared K-means, IGA and Nelder-Mead. Since the clustering results achieved by the IGA are satisfactory we have applied the IGA to the Image clustering problem by proposing a new variable length IGA (VLIGA) for automatic evolution of clusters. Experiments were carried out with three standard natural grey scale images to evaluate the performance of the proposed VLIGA. It was evident from the results that VLIGA algorithm was effective compared to the GCUK [12] and traditional K-means algorithm. Further enhancements will include the study of higher dimensional data sets and large data set for clustering. Also the datasets with mixed data can be studied. It is also planned to study the appropriateness of hybrid algorithm (K-NM-IGA) for image clustering and extend the same to color images.

References

[1] K.Jain, R.C.Dubes, " Algorithms for Clustering Data", Prentice-Hall, Englewood Cliffs, NJ,1988
[2] R.O Duda and P.E hart, "Pattern classification and Scene Analysis", John Willy & Sons, NY, USA, 1973
[3] J.T.Tou, R.C.Gonzalez "Pattern Recognition Principles", Addision-Wesley, Reading 1974
[4] D.E.Goldberg, "Genetic Algorithms in Search, Optimization and machine Learning", Addision-Wesley, New York, 1989..
[5] U.Maulick and S.Bandyopadhyay,"genetic Algorithm based Data Clustering Techniques", pattern recognition, 33(2000) 1455-1465..
[6] B.B.Mishra, S.C.Satapathy et al, " A fast and efficient Genetic Algorithm for Data Clustering", International Conference on Systemic, Cybernetics and Informatics PP-695-700, 2006.
[7] M.Srinivas and L.M. Pattanaik," Genetic Algorithm: A survey," IEEE Computer, vol. 27, pp. 17-27, June 1994
[8] Suresh Chandra Satapathy, Venkatesh Katari at.el: A Novel Approach for Integrating PSO and Improved GA for Clustering using Parallel and Transitional techniques.
[9] Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. Computer Journal
[10] Spendley, W., Hext, G. R., & Himsworth, F. R. (1962). Sequential application of simplex designs in optimization and evolutionary operation. Technometrics.
[11] Suresh Chandra Satapathy, Venkatesh Katari at.el: An Efficient Data Clustering Algorithm using Improved GA and Nelder-Mead Simplex Search. Accepeted in ICCIMA 2007, Sivakasi, India.
[12] Bandyopadhyay, S., and Maulik, U., 2002. Genetic clustering for automatic evolution of clusters and application to imageclassification. IEEE pattern recognition, Vol.35, p.1197-1208.
[13] Bandyopadhyay, S., and Maulik, U., 2001. Nonparametric genetic clustering: comparison of validity index. IEEE Transactions on systems man, and cybernetics-part C: Applications and reviews, 31(1), pp.120-125.
[14] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Patt. Anal. Mach. Intell. 1 (1979) 224–227.
[15] Yang, M.S., and Wu, K.L., 2001. A new validity index for fuzzy clustering. IEEE International Fuzzy Systems conference, pp.89-92.
[16] Ross, T.J., 1995. Fuzzy logic with engineering applications Mc Graw-hill, Singapore, 592p.

Venkatesh Katari : He is presently doing his final year in Computer Science Engineering in GITAM, Visakhapatnam. He published few IEEE conference papers on Data clustering and International conference papers on Robotics and Machine learning. His areas of interest include Datamining, Robotics & A.I.

Suresh Chandra Satapathy : He is a Professor in Computer Science Engg in ANITS, Vishakapatnam and doing his PhD in Data Mining using Swarm Intelligence techniques. He has published many IEEE conference papers on Data clustering using PSO, GA etc. His areas of interest include Data mining, machine learning, Swarm Intelligence etc.

JVR Murthy : H**e** is a Professor in Computer Science Engg in JNTU College of Engineering, kakinada. He has published many IEEE conference papers and journal papers on Data Mining and data warehousing. He is currently guiding many students on Data Mining for their PhD. His areas of interest include Data mining,, Data warehousing, Dimensional Modeling etc.

PVGD Prasad Reddy: He is a Professor in Computer Science Engg in AU Engineering College, Andhra University, Vishakapatnam. He has published many IEEE conference papers on Data Mining, Image Processing etc. He has guided several scholars for their PhDs. His areas of interest include Data mining, machine learning, Image Processing etc.