http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (4): 554-560, 2007
ISSN 1812-5638

Multispectral Image Clustering Using Enhanced Genetic k-Means Algorithm

[1]K. Venkatalakshmi, [1]P. Anisha Praisy, [1]R. Maragathavalli and [2]S. MercyShalinie
[1]Department of IT, Thiagarajar College of Engineering, Madurai-625015, Tamil Nadu, India
[2]Department of CSE, Thiagarajar College of Engineering, Madurai-625015, Tamil Nadu, India

Abstract: An attempt has been made in this study to find globally optimal cluster centers for multispectral images with Enhanced Genetic k-Means algorithm. The idea is to avoid the expensive crossover or fitness to produce valid clusters in pure GA and to improve the convergence time. The drawback of using pure GA in this problem is the usage of an expensive crossover or fitness to produce valid clusters (Non-empty clusters). To circumvent the disadvantage of GA, hybridization of GA with k-Means as Genetic k-Means is already proposed. This Genetic k-Means Algorithm (GKA) always finds the globally optimal cluster centers but the drawback is the usage of an expensive fitness function which involves σ truncation. The Enhanced GKA alleviates the problem by using a simple fitness function with an incremental factor. A k-Means operator (one-step of k-Means algorithm) used in GKA as a search operator is adopted in this study. In Enhanced GKA the mutation involves less computation than the mutation involved in GKA. In order to avoid the invalid clusters formed during the iterations the empty clusters are converted into singleton cluster by adding a randomly selected data item until none of the cluster is empty. The results show that the proposed algorithm converges to the global optimum in fewer numbers of generations than conventional GA and also found to consume less computational complexity than GKA. It proves to be an effective clustering algorithm for multispectral images.

Key words: Allele, clustering, genetic k-Means algorithm (GKA), global optimization, k-means operator, multispectral image, total within cluster variation

INTRODUCTION

Evolutionary algorithms are very good in search. They can even perform parallel searches, in complex search spaces. A popular evolutionary algorithm is Genetic algorithm, which is well known for its robustness even in large search spaces (Fogel *et al*., 1994). In this study an enhanced GKA, which is a modification of GA and an enhancement of GKA, is proposed.

Clustering has been used in various areas such as psychology, biology, medicine etc. Cluster analysis organizes the input into groups such that the items within the group are more similar to each other than the items belonging to different groups (Jain and Dubes, 1989). There are various types of clustering algorithms. They are mainly classified as partitional clustering algorithms and hierarchical clustering algorithms. In this study, partitional clustering has been given focus (Jones and Betranio, 1991). The objective here is to partition the data into specified number of clusters such that it minimizes the Total Within Cluster Variation (TWCV). TWCV is given by the sum of distances between all data item and their respective cluster center.

In general, partitional clustering algorithms like Hill-Climbing will settle at the local optimum. One of the iterative Hill-Climbing algorithms, k-Means which is well known for its simplicity also suffers from the above problem. Stochastic approaches like simulated annealing (Selim and Alsultan, 1991; Klein and Dubes, 1989), genetic algorithm are very good in finding the global optima but it takes time to converge to the global optima due to its expensive crossover. To obtain faster convergence as well as to retain the simplicity of the k-Means, the two algorithms have been combined (i.e.,) the crossover operator in conventional GA is replaced by k-Mean operator (one step k-Mean algorithm)) and this has been used in GKA (Babu and Murty, 1994).

The hybridization of GA with k-Means yields a greater benefit over conventional GA, but it uses an expensive fitness calculation. The fitness calculation involves the σ-truncation, which in turn consumes the calculation of average and standard deviation. As a result, there is an increase in computational complexity. The study overwhelms the problem by using a simple fitness calculation with the aid of an incremental factor. Enhanced GKA also replaces the mutation by a simple formula, which also satisfies the intended purpose of mutation. As a result this Enhanced GKA has an edge over the GKA.

Corresponding Author: K. Venkatalakshmi, Department of IT, Thiagarajar College of Engineering, Madurai-625015, Tamil Nadu, India Tel: (0452) 2482240/507

GENETIC k-MEANS ALGORITHM

The hybrid algorithm is formed by the fusion of GA a stochastic approach, with k-Means an attractive and an iterative algorithm (Krishna and Murty, 1999). Both algorithms are chosen for fusion, since the resulting algorithm overcomes individual obstacles found during convergence towards global optima. The resulting hybrid algorithm is called the genetic k-Means algorithm (GKA). The k-Means operator, one step of k-Means algorithm, is used in GKA instead of the crossover operator used in conventional GA's. In GKA a distance based mutation is used in which a randomly selected allele is flipped. Thus, GKA combines the simplicity of the k-Means algorithm and the robust nature of GA's. However the fitness function used is simpler than conventional GA, it is also still an overhead as it involved σ- truncation as a part of it, which in turn depends on the average and standard deviation. This study suggests a simple fitness function compared to the fitness function involved in GKA. Here the probability of mutation calculation is also simplified which in turn reduces the time needed for convergence. Thereby this process proves to be better than GKA in terms of computational complexity. It is found that Enhanced GKA always finds the global optima. It is found to converge to the global optimum with 100% accuracy and hence it is proved that Enhanced GKA has less computational complexity than the GKA.

ENHANCED GENETIC k-MEANS ALGORITHM

Similar to GKA and GA, Enhanced GKA also works on population of encoded string called as chromosome. The chromosomes are initialized randomly at the initial stage. Then performing the genetic operations on the chromosomes until termination condition is reached evolves the next generation. Various genetic operators, which are listed below, are acted on each chromosome in each generation.

Let there be n input data items and the objective is to find k cluster centers such that the cluster centers minimizes the TWCV.

Coding: A simplest and an understandable way of coding is to consider a chromosome of length n and allow each allele to have value from {1, 2…….k}. Here each allele refers to a data item corresponding to the position on the chromosome and its value represents the value of the cluster to which it belongs. This type of coding is called as string-of-group-numbers encoding.

Initialization: In the initial population the chromosome is assigned randomly such that each data point belongs to any one of the clusters. But however even when one of the cluster is empty illegal chromosomes may result (i.e., chromosome having empty clusters). To avoid empty clusters the data items are split into two parts with number of elements greater than or equal to k in first part so as to initialize at least one pixel to each of the k-clusters. The first part of the data item is equally distributed to every cluster. The other data part is initialized randomly to anyone of the clusters.

Fitness calculation: High fitness value is assigned to a chromosome with less Total Within Cluster Variation. Since a high value of fitness is assigned to a chromosome of less TWCV it is a minimization problem. To convert minimization problem into maximization problem the inverse of TWCV is considered as the fitness function.

$$f(i) = 1/\sum_{j=1}^{n} \begin{matrix}(\text{CON} + \text{difference of allele j in chromosome} \\ \text{i from the corresponding cluster centre})\end{matrix} \quad (1)$$

If the difference is small, then the chromosome will have a high fitness value. In Eq. 1 the CON is a constant whose value is 1. It is inserted to handle the exceptional cases where the TWCV is zero. If TWCV = 0 then in the absence of CON, fitness will be equal to infinity. The fitness formula used is less expensive in terms of computation than the fitness used in GKA. The usage of σ-truncation in GKA increases the computational complexity. But Eq. 1 does not involve such expensive computation. Since the fitness calculation is used in every generation and for every chromosome, the reduction in complexity of fitness function will have a greater impact on time complexity (Table 1).

Selection: The objective is to find the globally optimized cluster centers and thereby reduce the TWCV. Hence the chromosome with high fitness based on Eq. 1 is selected and placed on the mating pool. The chromosomes are sorted based on the fitness value and then the chromosome with worst fitness is truncated and the rest are retained. Finally the chromosome with best fitness value is duplicated to maintain the population size.

k-mean operator (KMO): Instead of genetic crossover operator, k-Mean Operator (one step of k-Mean algorithm) adopted from GKA is used, because crossover is expensive to produce valid chromosome (i.e., chromosome with nonempty clusters) and also leads to the formation of invalid chromosomes as shown in Fig. 1.

The following two steps constitute KMO:

Table 1: Fitness value calculated using Eq. 1

Generation No.	1	2	3	4	5	6	.	.	.	19	20
1	0.0016	0.0014	0.0015	0.0014	0.0014	0.0015	.	.	.	0.0018	0.0013
2	0.0025	0.0018	0.0025	0.0018	0.0024	0.0024	.	.	.	0.0018	0.0044
3	0.0012	0.0017	0.0029	0.0017	0.002	0.0027	.	.	.	0.0018	0.0044
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
100	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	.	.	.	0.0099	0.0099

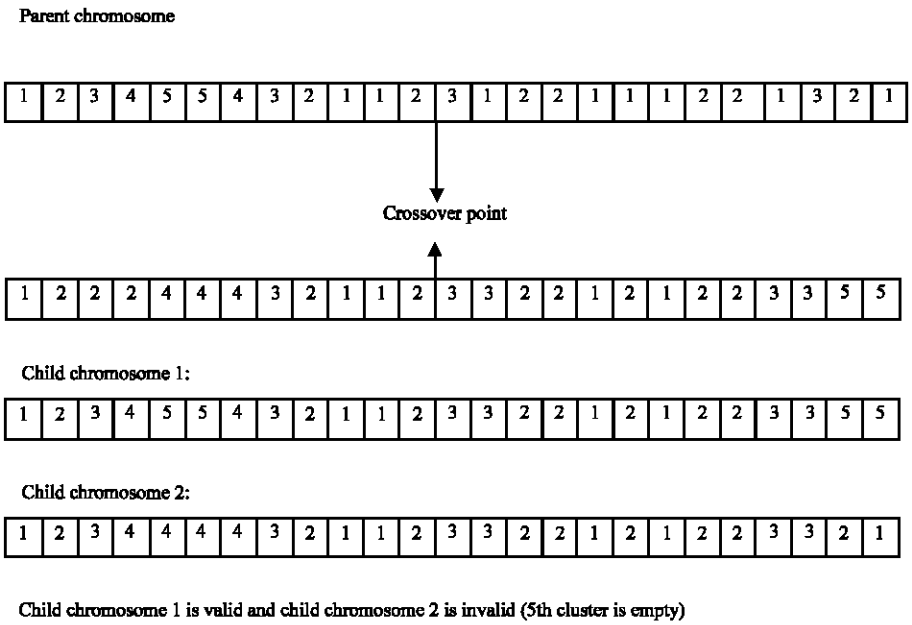

Fig. 1: Single point crossover

- The cluster center is calculated for every cluster by taking the average of data points inside that cluster.
- Reassign each data point to the cluster such that the distance from the cluster center to that corresponding data point is minimum compared to other cluster centers.

But the drawback of these simple operations is that it may result in empty clusters. To overcome this problem, whenever an empty cluster is found after KMO a random data point is just added to that empty cluster and it is removed from the cluster to which it actually belongs. It is iterated until none of the clusters are empty.

$$\text{Sum of Distance} = \sum_{j=1}^{K} |C_i - P_{ij}| \qquad (2)$$

Where I = {1, 2 ……n}

Here P_{ij} is the pixel j in ith class and C_i is the center of the ith class. The empty clusters are identified using Eq. 2. If the sum of distance is zero then it corresponds to empty cluster. The identification of empty clusters using Eq. 2 works well when the input data item contains non-duplicate values. But however if the input has duplicate values then the data point may overlap with cluster center and in such case the nonempty cluster may also resemble an empty cluster.

Mutation: The mutation is used to avoid local optimum and to make the cluster center to propagate toward the global optimum. Due to the random nature of initialization, cluster center will be improper during the initial stages and it requires at least 10 generations to be settled. It is performed only after 10th generation.

$$\begin{aligned} D_m(X_{ij}) &= (\text{abs}\ (X_{ij} - C_j) - \text{tot\_min}) \\ P_m(X_{ij}) &= 0 \text{ if } D_m = 0 \\ P_m(X_{ij}) &= 1 \text{ otherwise} \end{aligned} \qquad (3)$$

Where tot_min = minimum difference between the allele and the k cluster centers, P_m is the probability of mutation and D_m is the distance parameter.

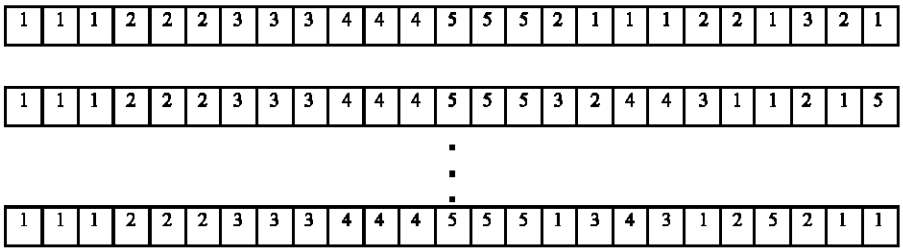

Fig. 2: Randomly initialized cluster

The probability of mutation is calculated according to above equation and if P_m is 1, then the allele value mapping to the cluster center is randomly initialized to some other cluster, which otherwise is left as such. As a result of mutation, empty clusters may form if mutation operation is done on a singleton cluster. To avoid the problem of formation of empty clusters, mutation is performed on an allele only if the sum of difference of all the data points and the corresponding cluster center of the mutated chromosome is non-zero. Otherwise another random allele is generated and it is mutated until a valid mutated child (all clusters are nonempty) is formed.

All the above steps are performed on the given data set. The steps from 3 to 6 are repeated over each generation until the termination condition is reached. Generally, there are two methods of termination. One method is to terminate if required number of generations are completed and another way is to stop when the cluster centers remain constant between iterations without any change. The former approach is used in this study. The complete workflow of the process used is shown in Fig. 2.

EXPERIMENTAL STUDY

For the experimental purpose of the Enhanced GKA, the following 5×5 array is considered, which easily explains the procedure.

$$\begin{bmatrix} 105 & 101 & 86 & 83 & 95 \\ 107 & 99 & 98 & 78 & 98 \\ 102 & 130 & 142 & 154 & 98 \\ 131 & 127 & 144 & 139 & 112 \\ 121 & 134 & 131 & 93 & 82 \end{bmatrix}$$

In this experiment, Enhanced GKA tries to find k globally optimum cluster centers such that it minimizes TWCV. Here the value of k is assigned as 5.

Coding: In this study, chromosome length is 25 which corresponds to the number of pixels in the above matrix. Here each allele corresponds to a pixel of the respective location and its value maps to the cluster to which it belongs. The 5 centers are randomly initialized to anyone of the 25 pixels shown above. These randomly generated clusters are made to converge towards the globally optimum cluster centers as a result of GKA. In this experiment the randomly generated centers are cen1 = 107, cen2 = 98, cen3 = 78, cen4 = 121, cen5 = 82. The population size is made equal to 20.

Initialization: In the initialization stage, population size is fixed. Here each allele is randomly initialized to anyone of the cluster. Due to the random nature of initialization, invalid chromosomes (non empty clusters) may result. To overwhelm the invalid chromosomes, 25 pixels are split into two parts of size 15 and 10. The former part of pixels is equally distributed into 5 clusters as shown in Fig. 2. As a consequence, none of the clusters remain empty. The latter part of the pixels is randomly distributed to anyone of the clusters. Thus as a result of the initialization stage, a fixed set (population size) of valid chromosome is generated.

Fitness evaluation: The Fitness of the chromosome represents how much it is fit to survive in the next generation. The fitness of all the chromosomes are calculated using Eq. 1 which is shown in Table 3. Then the fitness values are sorted. For illustration purpose, consider the iteration1 shown in Table 3. The chromosome 19 with the fitness value 0.0018 is the best chromosome and the chromosome 18 with fitness 0.0012 is the worst chromosome. The accuracy is lower in 1st generation due to initial random initialization. As generations evolve, the fitness value of the chromosome increases finally to 0.0099 in this experiment.

Selection: Based on fitness value evaluated the selection operation is performed such that Enhanced GKA retains the best chromosome. Consider the row 1 of Table 3, the chromosome 19 with fitness of 0.0018 corresponds to the best chromosome in generation 1 and the chromosome

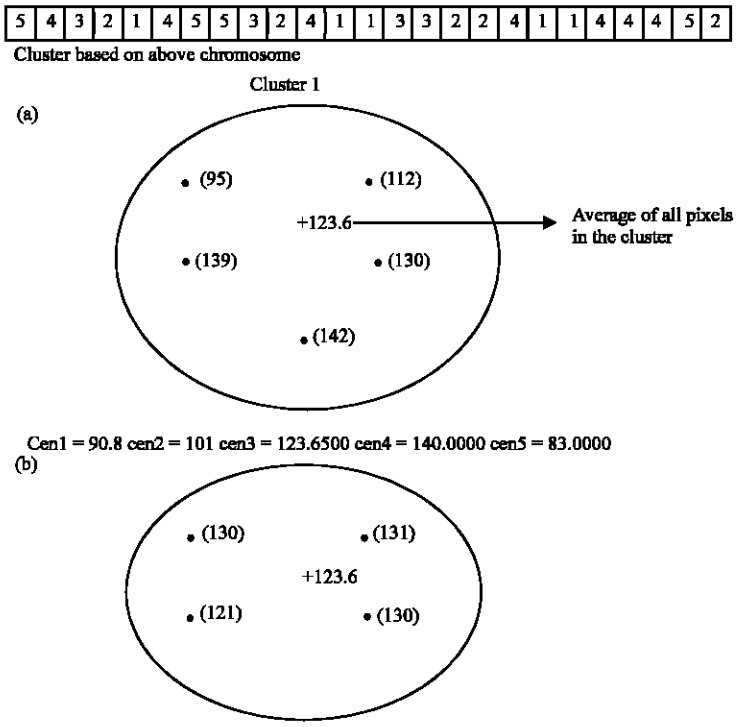

Fig. 3: k- mean operator

18 with the fitness of 0.0012 corresponds to worst chromosome. Based on the selection methodology adopted in this study the 18th chromosome is truncated and the rest of the chromosomes are retained. Finally in order to maintain the population size the chromosome 19 which is the best chromosome in generation 1 is duplicated. The selected chromosomes are placed in the mating pool.

k-mean operator (KMO): The KMO is applied to every chromosome placed on the mating pool. In the step 1 of this process using the value of the allele and its location on the chromosome, the pixels are categorized into respective clusters as shown in Fig. 3 and then new cluster center is formed for each cluster by computing average of the pixels. It is illustrated in Fig. 3a. Then based on cluster centers each pixel is reassigned to the nearest cluster center as shown in Fig. 3b. If empty clusters are formed then a random pixel in the range {1, 2…..25} is generated and added to the empty cluster, thereby converting empty cluster into singleton cluster.

Table 2: Cluster centers with mutation every 10 generation

Generation No.	Center 1	Center 2	Center 3	Center 4	Center 5
10	82.2500	97.4286	104.6667	116.5000	136.8889
20	82.2500	98.0000	108.0000	129.0000	144.7500
30	82.2500	98.0000	108.0000	129.0000	144.7500
40	82.2500	98.0000	106.0000	116.5000	136.8889
50	82.2500	99.6000	116.5000	130.6000	144.7500
60	82.2500	98.0000	108.0000	129.0000	144.7500
70	82.2500	98.0000	106.0000	116.5000	136.8889
80	82.2500	98.0000	106.0000	116.5000	136.8889
90	82.2500	99.6000	116.5000	130.6000	144.7500
100	82.2500	94.0000	100.1429	113.3333	136.8889

Table 3: Cluster centers without mutation every 10 generation

Generation No.	Center 1	Center 2	Center 3	Center 4	Center 5
10	82.2500	99.6000	116.5000	130.6000	144.7500
20	82.2500	98.0000	108.0000	129.0000	144.7500
30	82.2500	96.8333	103.7500	116.5000	136.88889
40	82.2500	97.4286	104.6667	116.5000	136.8889
50	82.2500	98.0000	106.0000	116.5000	136.8889
60	82.2500	98.7778	113.3333	132.0000	146.6667
70	82.2500	98.0000	108.0000	129.0000	144.7500
80	82.2500	94.0000	100.1429	113.3333	136.8889
90	82.2500	97.4286	104.6667	116.5000	136.8889
100	82.2500	98.0000	106.0000	116.5000	136.8889

Mutation: It is performed only after the tenth generation. The probability of mutation is calculated according to Eq. 3. The allele for mutation is selected by generating a random number between [1, 2…25]. Then the random allele is flipped to the value of another cluster center if $P_m = 1$, which otherwise is left as such. As a result of mutation, if empty cluster arises then mutation is not performed on that allele. Another random allele [1, 2… 25]

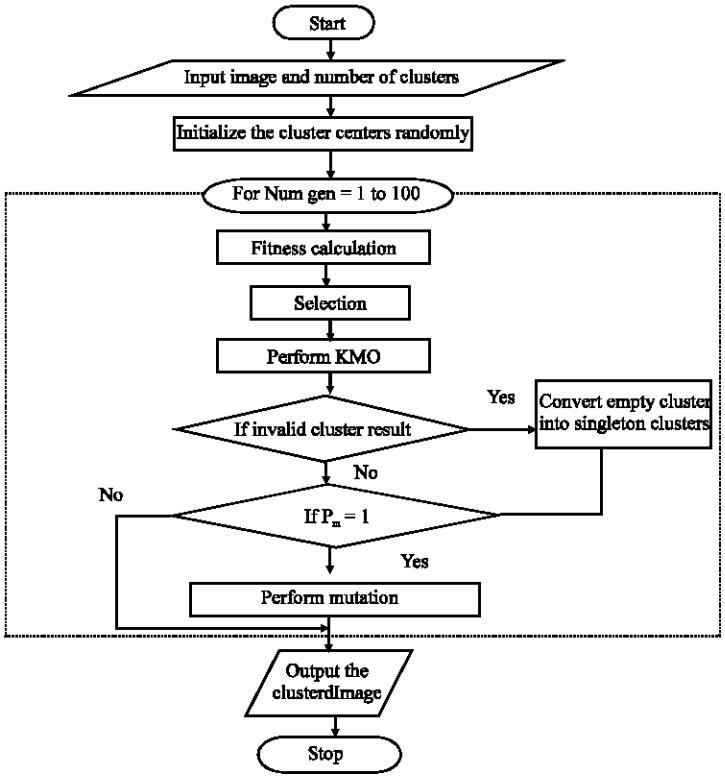

Fig. 4: Design flow of the clustering process

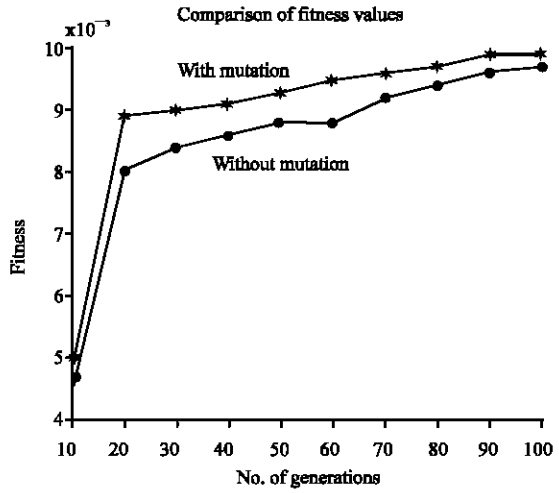

Fig. 5: Plot of fitness against number of generations

is tried for mutation. The plot of fitness against number of generations in (Fig. 5) reveals that mutation makes cluster centers to converge more quickly to global optima. The above steps are iterated for 100 generations. Then finally the Enhanced GKA was found to converge to the global optimum. The cluster centers for the input 5×5 matrix obtained are- 82.25, 94, 100.1429, 113.3333 and 136.889. The values of cluster center with and without mutation are listed in Table 2 and 3, respectively. A comparative plot of the fitness is showed in Fig. 5.

RESULTS AND DISCUSSION

The experimental data is a multispectral image consisting of three categories: vegetation area, water body and wasteland as shown in Fig. 7. Based on the above algorithm, three random values are generated and initialized as cluster centers. Then the above operators are applied in an iterative manner. The final cluster centers are found to converge to global optimum. The clustered image is shown in Fig. 8. It proves to have 100% accuracy with cluster centers 51, 103 and 182 and the time elapsed is 15.893 sec. A plot between number of generations and computational time in Fig. 6 shows that the enhanced GKA converges to global optimum in less time than GKA. Future research interest includes modifying the Eq. 2 used in Enhanced GKA to find the empty cluster. Since the equation is based on difference, the redundant data will overlap and pretend as an empty cluster.

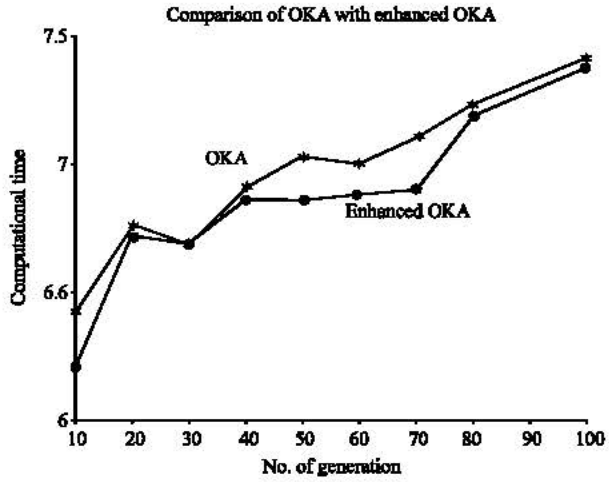

Fig. 6: Plot of number of generations vs computational time

Fig. 7: Original image

Fig. 8: Clustered image

REFERENCES

Babu, G.P. and M.N. Murty, 1993. A near-optimal initial seed selection in K-means algorithm using a genetic algorithm. Pattern Recognit. Lett., 14: 763-769.

Babu, G.P. and M.N. Murty, 1994. Simulated annealing for selecting initial seeds in the k-means algorithm. Ind. J. Pure Applied Math., 25: 85-94.

Fogel, D.B., 1994. An introduction to simulated evolutionary optimization. IEEE Trans. Neural Networks, 5: 3-14.

Jain, A.K. and Dubes, R.C. 1989. Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall.

Jones, D.R. and M.A. Beltramo, 1991. Solving partitioning problems with genetic algorithms. In: Proc. 4th Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufman.

Klein, R.W. and R.C. Dubes, 1989. Experiments in projection and clustering by simulated annealing. Pattern Recognit., 22: 213-220.

Krishna, K. and M. Murty, 1999. Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 29: 433-439.

Selim, S.Z. and K. Alsultan, 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognit., 10: 1003-1008.