Quasi-based hierarchical clustering for land cover mapping using satellite images

J. Senthilnath[a1], Ankur raj[b2], S.N. Omkar[a3], V. Mani[a4], Deepak kumar[c5]

[a]Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

[b]Department of Information technology, National Institute of Technology, Karnataka, India

[c]Department of Electrical Engineering, Indian Institute of Science, Bangalore, India

{[1]snrj@aero.iisc.ernet.in;[2]ankurrj7@gmail.com; [3]omkar@aero.iisc.ernet.in;
[4]mani@aero.iisc.ernet.in; [5]deepak@ee.iisc.ernet.in}

Abstract. This paper presents an improved hierarchical clustering algorithm for land cover mapping problem using quasi-random distribution. Initially, Niche Particle Swarm Optimization (NPSO) with pseudo/quasi-random distribution is used for splitting the data into number of cluster centers by satisfying Bayesian Information Criteria (BIC).The main objective is to search and locate the best possible number of cluster and its centers. NPSO which highly depends on the initial distribution of particles in search space is not been exploited to its full potential. In this study, we have compared more uniformly distributed quasi-random with pseudo-random distribution with NPSO for splitting data set. Here to generate quasi-random distribution, Faure method has been used. Performance of previously proposed methods namely K-means, Mean Shift Clustering (MSC) and NPSO with pseudo-random is compared with the proposed approach - NPSO with quasi distribution(Faure).These algorithms are used on synthetic data set and multi-spectral satellite image (Landsat 7 thematic mapper). From the result obtained we conclude that use of quasi-random sequence with NPSO for hierarchical clustering algorithm results in a more accurate data classification.

Keywords: Niche Particle Swarm Optimization, Faure sequence, Hierarchical clustering.

1 Introduction

Nature has given a lot to mankind and land is one such resource. We need actual information regarding the features of land to make good use of it. Using satellite images, we can accurately plan and use land efficiently. Satellite images offer a method of extracting this temporal data that can be used in gaining knowledge

J. C. Bansal et al. (eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012),* Advances in Intelligent Systems and Computing 202, DOI: 10.1007/978-81-322-1041-2_5, © Springer India 2013

regarding land use. Recent advances in the realm of computer science has allowed us perform “intelligent” jobs. This has established a vast research area in solving the automatic image clustering problem. The image clustering using satellite image for land cover mapping problem is useful for auditing the land-usage and city planning [1].

The main objective of clustering problem is to minimize the intra-cluster distance and maximize the inter-cluster distance [2]. One of the main task of clustering problem is to locate the cluster centres for a given data set; it is basically a problem of locating maxima of a mapped function from a discrete data set. Recently researchers are interested in capturing multiple local optima of a given multi-modal function for this purpose nature inspired algorithms are used. Brits *et.al* [3] developed Niche PSO (NPSO) for optimization of standard benchmark functions, later Senthilnath *et.al* [2] applied the same concept for locating multiple centres of a data set for hierarchical clustering problem.

NPSO is a population based algorithm its performance has shown high dependency on initial distribution of population in search space it has been observed in literatures that performance of particle swarm optimization has improved by using more uniform distribution of particle in search space [4]. Kimura *et.al* [5] have used Halton sequence for initializing the population for Genetic Algorithms (GA) and have shown that a real coded GA performs much better when initialized with a quasi-random sequence in comparison to a GA which is initialized with a population having uniform probability distribution (i.e. pseudo-random distribution). Instances where quasi-random sequences have been used for initializing the swarm in PSO can be found in [4, 5, 6, 7]. Nguyen *et.al* [7] has given a detailed comparison of Halton, Faure and Sobol sequences for initializing the swarm. It has been observed that performance of Faure sequence takes over the performance of Halton sequence in terms of uniformity in space.

In this paper, the comparison is done between pseudo and quasi based distribution for initializing NPSO to capture multiple local maxima for a given data set. In our study the data set used are synthetic data set and Landsat satellite image for hierarchical clustering. In earlier studies [4, 5, 6, 7] for optimization problem, it has been observed that use of quasi sequence for initializing population in PSO has given a better performance. The same approach has been applied in this study using the quasi sequence with NPSO for hierarchical clustering algorithm. NPSO is used to split complex large data set into number of cluster satisfying Bayesian Information criteria (BIC) which is commonly used in model selection [8]. These cluster centres are used for merging the data set to their respective group. The challenge is how to get better classification efficiency using quasi distribution in NPSO with hierarchical clustering algorithm.

2 Random Sequences

The clustering using population based methods require initial random distribution of points to extract optimal cluster centres. To generate truly random numbers there is a requirement of precise, accurate, and repeatable system measurements of absolutely non-deterministic processes. Computers normally cannot generate truly random numbers, but frequently are used to generate sequences of pseudo-random numbers. There are two principal methods used to generate random numbers. One measures some physical phenomenon that is expected to be random and then compensates for possible biases in the measurement process. The other uses computational algorithms that produce long sequences of apparently random numbers called pseudo-random. It may be possible to find a more uniform distribution using low-discrepancy sequence known as quasi-random numbers. Such sequences have a definite pattern that fills in gaps evenly, whereas pseudo-random sequence unevenly distributes the sequence, this leads to larger gaps in search space.

2.1 Pseudo-random sequences

A pseudo-random process is a process that appears to be random but is not. Pseudo-random sequences typically exhibit statistically randomness while being generated by an entirely deterministic casual process. These are generated by some algorithm, but appear for all practical purposes to be random. Random numbers are used in many applications, including population based method involving distribution of initial points using random numbers (pseudo number). A common pseudo-random number generation technique is called the linear *congruential* method [9]. The pseudo-random numbers are generated using following equation.

$$A_{n+1}=(Z * A_n + I) \bmod M \qquad 1$$

where A_n is the previous pseudo number generated, Z is a constant multiplier, I is a constant increment, and M is a constant modulus. For example, suppose Z is 7, I is 5, and M is 12 if the first random number (usually called the *seed*) A_0 is 4, then next pseudo number $A_1= (7*4+5) mod\ 12=9$. In this way we can generate the pseudo-random sequence.

2.2 Quasi random sequence

The quasi-random numbers have the low-discrepancy (LD) property that is a measure of uniformity for the distribution of the point mainly for the multi-dimensional case. The main advantage of quasi-random sequence in comparison to pseudo-random sequence is it distributes evenly hence there is no larger gaps and no cluster formation, this leads to spread the number over the entire region. The concept of LD is associated with the property that the successive numbers are added in a position as away as possible from the other numbers that is, avoiding clustering (grouping of numbers close to each other). The sequence is constructed based on some pattern such that each point is separated from the others, this leads to maximal separation between the points. This process takes care of evenly distribution random numbers in the entire search space [10, 11].

The most fundamental LD sequence for one dimension is generated by Van der corput method, further to continue random sequence in higher dimension Faure and Halton methods are used.

2.2.1 Faure sequence

Faure sequence is a method to generate LD sequence; it extends the idea of Van der corput sequence in higher dimension. The most basic way to generate quasi-random sequence is Van der corput method**,** this method uses two basic equations eq.2and eq.3 to transform a number *n* in 1-dimensional space with base *b* .

$$n = \sum_{j=0}^{m} a_j(n) b^j \qquad 2$$

$$\phi(n) = \sum_{j=0}^{m} a_j(n) b^{-j-1} \qquad 3$$

where *m* is the lowest integer that makes *$a_j(n)$ as 0* for all *j* > *m***.** Above equations are used at base *b*, which is a prime number in respective dimension.The Van der corput sequence, for the number *n* and base *b*, is generated by a three step procedure:

Step-1: The decimal base number *n* is expanded in the base *b* using eq.2

$$3 = 0 * 2^0 + 0 * 2^1 + 1 * 2^2 = 011$$

Step-2: The number in base *b* is reflected. In this example it is 011 is reflected 110

Step-3: Now the reflected number is written as fraction less than one using eq.3 writing 011 gives $\phi(3) = 1 * 2^{-0-1} + 1 * 2^{-1-1} + 0 * 2^{-2-1} = \frac{3}{4}$

Now let us consider *n*=4 length of sequence to be generated, and let n_1=1, n_2=2,n_3=3 and n_4=4 then quasi-random sequence in 1-dimensional space will be generated as follow.

For n_1=1, using eq.2

$n_1 = 1 * 2^0 + 0 * 2^1 + 0 * 2^2$ here$a_0 = 1$, $a_1 = 0$, $a_2 = 0$

Now using eq.3

$$\phi(n_1) = a_0 * 2^{-0-1} + a_1 * 2^{-1-1} + a_2 * 2^{-2-1} = \frac{1}{2}$$

Similarly calculating for 2 and 4 gives $\frac{1}{4}$ and $\frac{1}{8}$ respectively, hence the first 4 numbers of Van der corput sequence are $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}$

This is basic LD sequence in one dimension, for higher dimension LD sequences are generated using Halton and Faure method. In Halton method the sequence numbers are generated using different prime base for each *k*-dimension. For k^{th}-dimension the N^{th} number of the sequence is obtained by

$\phi(N, b_1), \phi(N, b_2)$...... $\phi(N, b_k)$ where *i=1...k*, b_i is the prime number greater than or equal to *i*.

Faure sequence is similar to Halton sequence with two major differences: Faure uses same base for all dimension and vector elements are permutated in higher dimension. For dimension one it uses Van der corput sequence to generate sequence for higher dimensions vector permutation is carried out using eq.4.

$$a_j^d(n) = \left(\sum_{j \geq i}^{m} \frac{j!}{i!(j-i)!} a^{d-1}(n) \right) \bmod b \qquad 4$$

where ${}^{j}C_i = \frac{j!}{j!(j-1)!}$

The base of a Faure sequence is the smallest prime number which is greater than or equal to the number of dimensions in the problem, say for one dimensional problem base 2 is taken. The sequence number is selected between [0,1), the quantity of number generated to complete a cycle increases as the number of dimension increases. For e.g. in base two, for a cycle two numbers are picked within an interval [0,1) i.e. for the first cycle (0,1/2) and for second cycle (1/4,3/4) are selected, similarly for base 3 in the first cycle (0,1/3,2/3) are picked and for second cycle (1/9,4/9,7/9) are selected, hence long cycles have the problem of higher computational time. As Halton sequence uses different bases in each dimension so it has a problem of long cycle length, but in case of Faure this problem has been reduced by taking same base for each dimension. By reordering the sequence within each dimension, a Faure sequence prevents some problems of correlation for high-dimensions, whereas Halton sequence fails to minimize the correlation [12].

2.2.2 Illustration of Faure sequence

Let us consider the same example as discussed in section 2.2.1. The Faure sequence in 1st dimension corresponding to first 4 numbers (n_1=1, n_2 =2, n_3=3 and n_4=4) will be same as that of Van der corput sequence i.e. 1/2, 1/4, 3/4 and 1/8 now numbers for second dimension using Faure Method will be calculated as follow:

For n_1=1 representing at base *b=2*, $a_2a_1a_0$=001, now using eq.4 for vector permutation

$$\textit{for}\ a_0^2(1)\, i = 0\,, \quad a_0 = ({}^{0}c_0a_0 + {}^{1}c_0a_1 + {}^{2}c_0a_2)\bmod b = (1+0+0)\bmod 2 = 1$$

$$\textit{for}\ a_1^2(1)\, i = 1 \quad a_1 = ({}^{1}c_1a_1 + {}^{2}c_1a_2)\bmod b = (0+0)\bmod 2 = 0$$

$$\textit{for}\ a_2^2(1)\, i = 2 \quad a_2 = ({}^{2}c_2a_2)\bmod b = 0 \bmod 2 = 0$$

$$\textit{for}\ n_2 = 2\ \textit{at base}\ b\ a_2a_1a_0 = 010\ \textit{now applying equation}\ 4$$

$$a_0^2(2) = ({}^{0}c_0a_0 + {}^{1}c_0a_1 + {}^{2}c_0a_2)\bmod b = (0+1+0)\bmod 2 = 1$$

$$a_1^2(2) = ({}^{1}c_1a_1 + {}^{2}c_1a_2)\bmod b = (1+0)\bmod 2 = 1$$

$$a_2^2(2) = ({}^{2}c_2a_2)\bmod 2 = 0 \bmod 2 = 0$$

Now applying eq.3we get

$\phi(1) = a_0^2(1) \times 2^{(-0-1)} + a_1^2(1) \times 2^{(-1-1)} + a_2^2(1) \times 2^{(-2-1)} = \frac{1}{2}$ and

$$\phi(2)= a_0^2(2)\times 2^{(-0-1)} + a_1^2(2)\times 2^{(-1-1)} + a_2^2(2)\times 2^{(-2-1)} = \frac{3}{4}$$

Similarly other numbers are generated. The first four numbers of Faure sequence in 2-dimension are (1/2, 1/2), (1/4,3/4), (3/4,1/4), (1/8,5/8).

2.2.3 Quasi and Pseudo distribution

Fig-1 shows the distribution of 100 particle in the search space of [-2,2]. Two dimensional Faure sequence has been taken for quasi-random number. It can be seen that in Fig-1a quasi sequence is very uniformly distributed in space (each grid has at-least one point) whereas pseudo sequence as shown in Fig-1b which is generated by matlab *random number generator (rand()* function) is not very uniform.

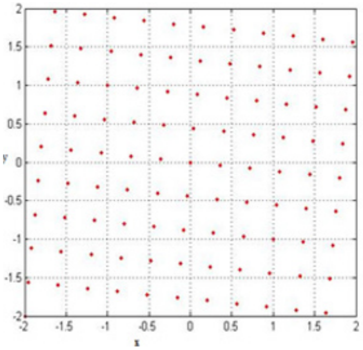

Fig-1a:Quasi-random distribution

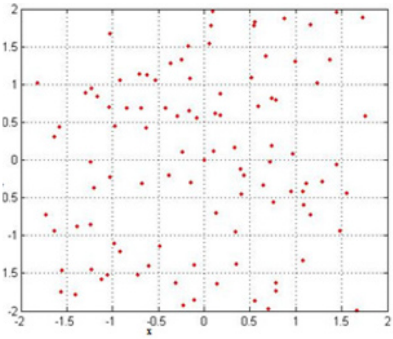

Fig-1b:Pseudo-random distribution

3 Cluster splitting and merging

The cluster analysis forms the assignment of data set into cluster based on some similarity measures. In this study an attempt to improve the performance of previously proposed hierarchical clustering is compared with quasi-random distribution with NPSO. The hierarchical splitting technique uses Kernel function for mapping discrete data set to an objective function. This is done by using a Gaussian Kernel, based on Euclidian distance between two data points *(r)* which is given by [13]

$$K(r) = e^{-\|r\|^2} \qquad 5$$

It is very difficult to predict how number of clusters is optimal for a given data set, as this is dependent on data distribution. A platform is provided using Bayesian Information criteria (BIC) which is a model fitting approach that provides an optimal number of clusters. The splitting of data set using BIC into number of cluster is given by [8, 15].

$$BIC \approx L(\theta) - \frac{1}{2} \times k_j \times \log(n) \qquad 6$$

where *L(θ)* is log-like hood measure, k_j is number of free parameters for specific number of cluster and *n* is no of data point for a given data set.

Niching techniques are modelled after a phenomenon in nature where animal species specialize in exploration and exploitation of different kinds of resources. The introduction of this specialization, or Niching, in a search algorithm allows it to divide the space in different areas and search them in parallel. The technique has proven useful when the problem domain includes multiple global and local optimal solutions. Brits *et. al* [3] implemented Niche particle swarm optimization (NPSO) which is a variant of PSO [14], based on flock of birds aimed to capture multiple optima in a multi-modal function.

The objective function of all the particles is calculated using Kernel function, using eq.5, if the variance in objective function value of the particle for some fixed number of iteration is less than some threshold value *ε* then it is named as sub-swarm leader.

The swarm is divided into several overlapping sub-swarms in order to detect multiple peaks. Sub-swarms are created with all particles around the local centre within the swarm radius. These particles are made to converge towards the local best position i.e. sub-swarm leaders

$$v_{i,j}(t+1) = wv_{i,j}(t) + \rho(t)\left(\hat{y}(t) - x_{i,j}(t)\right) \qquad 7$$

$$x_{i,j}(t+1) = -x_{i,j}(t) + v_{i,j}(t+1) \qquad 8$$

where $-x_{i,j}(t)$ resets the particle position towards the local best position, $\hat{y}_{i,j}(t)$ within sub-swarm radius, $w{*}v_{i,j}$ is the search direction, and *ρ(t)* is the region for the better solution. The personal best position of particle is updated using eq.10 where *f* denotes the objective function.

$$y_i(t+1) = \begin{cases} x_i(t+1) & if \quad f(x_i(t+1)) \le f(y_i(t)) \\ y_i(t) & if \quad f(x_i(t+1)) \ge f(y_i(t)) \end{cases} \qquad 9$$

The cluster centres generated using NPSO is grouped using agglomerative approach. These cluster centres are used for initializing K-means to perform agglomerative clustering [16, 17, 18]. Here parametric method is used to group the data points to the closest centres using similarity metric.

Merging data set algorithm:

Step-1: Results obtained as cluster centres from NPSO is given to K-means clustering.

Step-2: Merge data points to closest centres.

Step-3: Use voting method for each data points in the cluster.

Step-4: Cluster is grouped agglomerative using labels.

Step-5: Assign each data points to one of the class.

4 Results and discussion

In this section, we discuss the cluster splitting and merging by comparing pseudo and quasi-random distribution. This distribution is assigned initially for *n* particles in NPSO. We evaluate the performance of NPSO on synthetic and satellite data sets using the classification matrix of size *n* x *n*, where *n* is the number of classes. A value $A_{i,j}$ in this matrix indicates the number of samples of class *i* which have been classified into class *j*. For an ideal classifier, the classification matrix is diagonal. However, we get off-diagonal elements due to misclassification. Accuracy assessment of these classification techniques is done using individual (η_i), average (η_a) and overall efficiency (η_o) [15, 19].

4.1 Synthetic data set

The above algorithm is been applied for classification of a synthetic data set, the original data set consists of two classes, in each class there are 500 samples as shown in Fig-2a. The BIC analysis is carried out as shown in Fig-2b, the optimal clusters for this data set is 8. The hierarchical clustering technique using NPSO is used to generate the cluster centres by initializing the population based on pseudo and quasi-random distribution, in quasi-random Faure method is used.

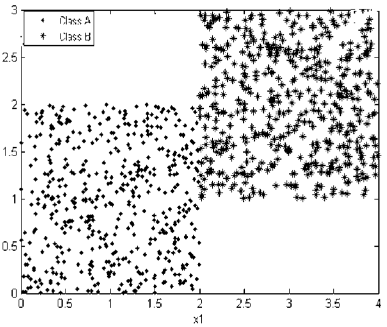

Fig-2a: Synthetic data set

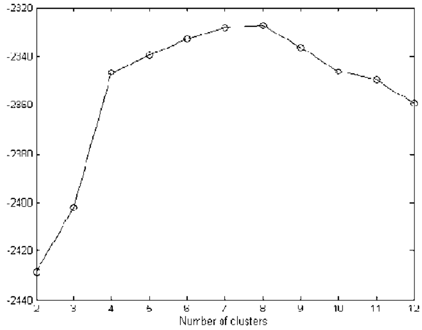

Fig-2b: BIC for synthetic data set

In NPSO, to set the parameter value for sub-swarm radius, inertia weight and weight of leader follower (ρ) different runs are carried out. From Fig-3 we can observe that the optimal parameter value for weight of leader follower (ρ) is 0.4. As it can be observed from Fig-4 that using quasi-random distribution more uniform variation of number of clusters with weight of leader follower (ρ), hence it is easy to predict the parameter value. In contrast for pseudo-random distribution variation is very high which makes difficult to initialize the parameter. The other parameter values assigned are: number of population is 300, sub-swarm radius is 0.1 and inertia weight is adaptively varied in interval [0.6, 0.7]. The 8 cluster centres obtained from NPSO is merged to obtain exact number of classes using merging technique as discussed above. The classification matrix obtained after merging using NPSO based on quasi-random is as shown in table 1.

Also the same experiment is repeated using NPSO based on pseudo-random distribution by keeping all the parameter to be same. The classification matrix obtained by NPSO pseudo-random to split the clusters and merging the data set to their class labels is as shown in table 2.

From table 1 and table 2 we can compare that NPSO quasi-random distribution performed better for all the performance measures to that of NPSO pseudo-random distribution.

Table 1: NPSO-quasi based classification

DATA Class	Class-1	Class-2	Individual efficiency
Class-1 (η_1)	500	0	100%
Class-2 (η_2)	27	473	94.6%
OE (η_o)	97.3	AE (η_a)	97.3

Table 2: NPSO-pseudo based classification

Data class	Class-1	Class-2	Individual Efficiency
class-1 (η_1)	492	8	98%
class-2 (η_2)	50	450	90%
OE (η_o)	94%	AE (n_a)	94.2%

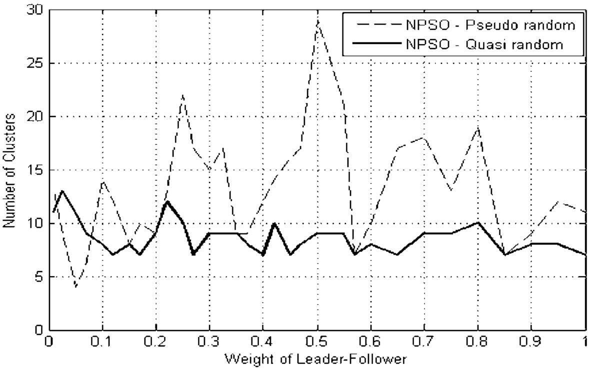

Fig-3: Effect of weight of leader follower in NPSO with quasi and pseudo distribution respectively

4.2 Landsat image

In this study, the Landsat image used is 15 X 15.75 Km^2 (500 X 525 pixels) and has 30m spatial resolution. The aim is to classify 9 land cover region using Landsat image. Senthilnath *et. al*[15] provides a detail description of data set. There are 9 level-2 land cover region for this image which include deciduous(C_1), deciduous pine(C_2), pine(C_3), water(C_4), agriculture(C_5), bareground(C_6), grass(C_7), urban (C_8) and shadow(C_9).

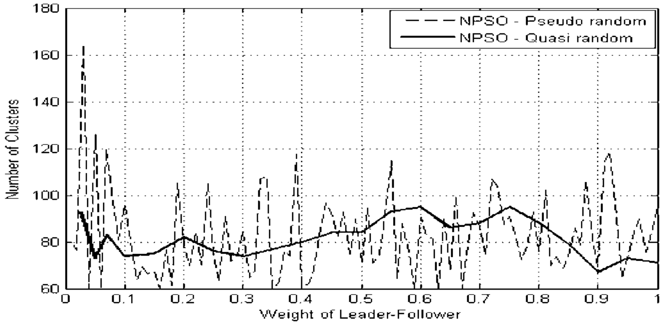

Fig-4 : Effect of weight of leader follower in NPSO with FAURE distribution In LANDSAT IMAGE

Table 3: Performance measure for K-means, MSC, and NPSO using Landsat data

Classification Efficiency	K-means**[2]**	MSC**[2]**	NPSO**[2]** (Pseudo)	NPSO (Faure)
η_1	82.1	85.9	85.0	90.39
η_2	68.0	81.0	81.3	88.78
η_3	53.9	69.3	82.6	90.99
η_4	92.3	92.4	92.4	94.93
η_5	76.3	76.2	77.9	80.46
η_6	35.6	38.6	70.7	84.35
η_7	67.3	69.2	72.2	79.76
η_8	39.9	41.7	70.8	81.84
η_9	28.4	66.8	77.8	81.16
η_a	60.4	69.0	78.9	85.85
η_o	70.8	78.1	81.8	88.14

Maximum cluster centres generated based on BIC for this data set should be 80 [2]. For this data set the NPSO parameter value assigned are: number of population is 500, sub-swarm radius is 0.1, inertia weight is adaptively varied in the interval [0.7,0.6], and weight of leader follower (ρ) is equal to 0.4. Among these parameter weight of leader follower plays an important role to generate the 80 cluster centres. The weight of leader follower for NPSO *(ρ)* was observed as most dominant factor, Fig-4 shows the variation of number of cluster centres generated with *(ρ)*, using pseudo-random distribution abrupt variation is observed due to high degree of randomness, whereas when Faure sequence is used as initial distribution as expected a more smooth curve is obtained. These cluster centres obtained from NPSO is merged to obtain exact number of classes using merging technique as discussed in section 3. The classification matrix obtained after merging is as shown in table 3.

From Table 3 we can observe that performance measure in all aspect using NPSO with Faure distribution based hierarchical clustering and classification is better in comparison to NPSO with pseudo based clustering for Landsat data. Fig-5 shows the classification results obtained for Landsat image using NPSO with Faure distribution.

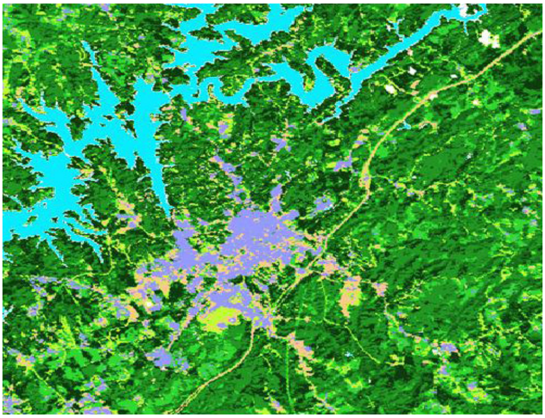

Fig-5: Classification using NPSO with Faure distribution

5 Conclusions and discussion

In this paper, we have presented an improved hierarchical clustering algorithm based on quasi-random distribution using NPSO. Initially NPSO with pseudo and quasi-random distribution is used to initialize the search space to split the data set into cluster centres by satisfying BIC. Here to generate the quasi-random sequence Faure method is used. Since by using the quasi-random sequence particles are distributed in the search space more uniformly, resulting in the accurate convergence of the particle to the centres.

An effect of weight of leader follower parameter has been analysed, it is observed that using quasi-random sequence to initialize weight of random component parameter of NPSO minimizes the random behaviour of the algorithm. This is useful to select a weight of leader follower value more accurately. The performance is measured using classification efficiency - individual, average and overall of the proposed algorithm. We observed that use of quasi-random sequence as initial distribution with NPSO results in better efficiency of classification for synthetic and Landsat data set.

References

[1] David, L.: Hyperspectral image data analysis as a high dimensional signal processing problem. IEEE Signal processing Mag. 19 (1), 17–28 (2002)

[2] Senthilnath, J., Omkar, S.N., Mani, V., Tejovanth, N., Diwakar, P.G., Shenoy, A.B.: Hierarchical clustering algorithm for land cover mapping using satellite images. IEEE journal of selected topics in applied earth observations and remote sensing. 5 (3), 762-768 (2012)

[3] Brits, R., Engelbrecht, A.P., van den Bergh, F.: A niching Particle Swarm Optimizer. In proceedings of the fourth Asia Pacific Conference on Simulated Evolution and learning. 692 –696 (2002)

[4] Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization in noisy and continuously changing environments. in Proceedings of International Conference on Artificial Intelligence and soft computing. 289-294 (2002)

[5] Kimura, S., Matsumura, K.: Genetic Algorithms using low discrepancy sequences. in proc of GEECO. 1341 –1346 (2005)

[6] Brits, R., Engelbrecht, A.P., van den Bergh, F.: Solving systems of unconstrained equations using particle swarm optimization. in proceedings of the IEEE Conference on Systems. Man and Cybernetics. 3, 102 – 107 (2002)

[7] Nguyen, X.H., Mckay, R.I., Tuan, P.M.: Initializing PSO with Randomized Low-Discrepancy Sequences: The Comparative Results, In Proc. of IEEE Congress on Evolutionary Algorithms. 1985 – 1992 (2007)

[8] Schwarz, G.: Estimating the dimension of a model. the Annals of statistics. 6 (2), 461-464 (1978)

[9] Donald, K.: Chapter 3 – Random Numbers". The Art of Computer Programming. Seminumerical algorithms (3 ed.) (1997)

[10] Niederreiter, H.: Quasi-Monte Carlo Methods and Pseudo Random Numbers. Bulletin of American Mathematical Society. 84(6) 957-1041 (1978)

[11] Marco A.G.D.: Quasi-Monte Carlo Simulation. http://www.puc-rio.br/marco.ind/quasi_mc2.html

[12] Galanti, S., Jung, A.: Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices. Journal of Derivatives. 63-83 (1997)

[13] Comaniciu, D., Meer, P.: Mean shift :a robust approach towards feature space analysis. IEEE Trans .pattern Anal .machIntell. 24 (5), 603-619 (2002)

[14] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, IV (Piscataway, NJ), IEEE Service Center. 1942–1948 (1995)

[15] Senthilnath, J., Omkar, S.N., Mani, V., Tejovanth, N., Diwakar, P.G., Archana, S.B.: Multi-spectral satellite image classification using glowwarm swarm optimization. in proc. IEEE int. Geoscience and Remote Sensing Symp (IGARSS).47-50 (2011)

[16] Li, H., Zang, K., Jiang ,T.: The regularized EM algorithm. in proc.20thNat.conf.Artificial Intelligence. 807-8 (2005)

[17] MacQueen ,J.: Some methods for classification and analysis of multi-variate observations. in proc .5th BerkeleySymp. 281-297 (1967)

[18] Senthilnath, J., Omkar, S. N., Mani, V.: Clustering using firefly algorithm – Performance study. Swarm and Evolutionary Computation. 1 (3), 164-171 (2011)

[19] Suresh, S., Sundararajan, N., Saratchandran, P.: A sequential multi-category classifier using radial basis function networks. Neurocomputing. 71, 1345-1358 (2008)