A Customized Particle Swarm Optimization for Classification of Multispectral Imagery Based on Feature Fusion

Venkatalakshmi Krishnan[1], Anisha Praisy[1], Maragathavalli R.[1], and Mercy Shalinie[2]
[1]Department of IT, Thiagarajar College of Engineering, India.
[2]Department of CSE, Thiagarajar College of Engineering, India.

Abstract: *An attempt has been made in this paper to classify multispectral images using customized particle swam optimization. To reduce the time consumption due to increase in dimensionality of multispectral imagery a preprocessing is done using feature extraction based on decision boundary. The customized particle swam optimization then works on the reduced multispectral imagery to find globally optimal cluster centers. Here particle swam optimization is tailored for classification of multispectral images as customized particle swam optimization. The modifications are performed on the velocity function such that velocity in each iteration is updated as a factor of g-best (global best) alone and the particle structure is made to incorporate the entire cluster centers of the reduced imagery. The initialization of particles is accomplished using modified k-means in order to retain the simplicity. AVIRIS images are used as test site and it was found that the customized particle swam optimization finds the globally optimal clusters with 98.56% accuracy.*

Keywords: *Multispectral image, decision based feature extraction, particle swam optimization, global optima, g-best.*

Received January 8, 2007; accepted May 3, 2007

1. Introduction

Classification of remote sensed data using evolutionary algorithms like neural network, genetic algorithm, Particle Swam Optimization (PSO), [3], and [l] has been under intensive investigation in recent years. One of the evolutionary algorithm PSO, shares many similarities with another evolutionary computation technique Genetic Algorithm (GA) [4]. The system is initialized with population of random solutions and searches for optima by updating generations. However unlike GA, PSO has no evolution operator such as crossover and mutation. In PSO the potential solutions called particles fly through the problem space by following the current optimum particles.

PSO is altered here as customized PSO to find the globally optimal cluster centers of the entire multispectral image set. The Particle structure adapted here is made to incorporate the details of cluster centers of the entire multispectral imagery set. The particle structure is of form P_0, P_1, P_2, P_3… P_{N*k} where the first k values stands for first image and $k+1$ to $2*k$ stands for second image and so on. In general the $(N-1)*k+1$ to $N*k$ stands for the cluster centers of N^{th} image. Thus each particle is of length $N*k$ where N stands for number of images in the multispectral imagery and k stands for number of globally optimal cluster centers. In order to retain the advantages of the statistical clustering methods the Modified *k*-Means algorithm [8] is used to initialize the particle structure.

The Modified *k*-Means algorithm is run on each image and *k* cluster centers are found and it is used to initialize each particle. The fitness of each particle is found and based on it the particle best and global best is found. Here the velocity is updated after each iteration based on the global best.

In this paper, in order to reduce the dimension of multispectral imagery the Decision Boundary based Feature Extraction method (DBFE) [5] is used. Because multispectral sensors may show quite different behaviors on observations even for same type of objects extraction of informative features for classification and elimination of redundant features is done using DBFE method. The classification fusion procedure used here greatly reduces the false classification rate. The customized PSO works on the reduced imagery of the DBFE.

2. Decision Boundary Feature Extraction

Feature extraction works well in reduction of dimensionality of massive datasets which is hard to manage. When hybrid classifiers are applied, feature extraction can be very important especially for multispectral images. Feature extraction can be viewed as finding the set of vectors that represent the observation while reducing the dimensionality. Here we focus on decision boundary feature extraction, which is based on extracting the features from the decision boundaries.

The DBFE focuses on classification accuracy rather than a surrogate to it such as using a statistical distance. DBFE functions on both mean separation and covariance distances, but some other feature extraction fail if there is no mean separation. The DBFE shows how many features are needed to achieve full accuracy. By looking at the eigen function generated, it provides some evidence as to which original features are the most important. The DBFE procedure can be returned in the following manner for two gaussian classes in order to determine the transformation needed to find the desired feature set:

- The *N* images of a particular imagery are fed as input and the number of the clusters is initialized as 2 just for the sake of simplicity.
- The Modified *k*-Means algorithm is applied for 10-20 iterations and clusters are found for the entire multispectral imagery. This step is performed to maintain the simplicity and at the same time to attain a more or less reasonable clusters.
- The mean and covariance for each of the two clusters are computed and named as M_i and $\sum_i$. Apply a chi-squared threshold test to the vectors of each cluster named as *X* which represents the pixels in the multispectral imagery and detect the outliers, i.e., for class W_i, retain *X* only if

  $$(X- Mi)\ t \sum 1\ \text{-}1\ (X- Mi) < Rt1 \quad (1)$$

  Here, let (X_1, $X_2 \ldots X_{n1}$) be the samples corresponding to first class and *n1* corresponds to the number of pixels in first class. R_{t1} is set as such the outliers are omitted and the correctly classified samples of first class are retained. In the rest of the steps only those samples which pass the above equation are used.
- Apply a chi-squared threshold test of first class to the samples of the second class and retain *Y* only if

  $$(Y- Mi)\ t \sum 2\ \text{-}1\ (Y- Mi) < Rt2 \quad (2)$$

  Here, let (Y_1, $Y_2 \ldots Y_{n2}$) be the samples corresponding to second class and *n2* corresponds to the number of pixels of second class. R_{t2} is set as such the outliers are omitted and the correctly classified samples of first class are retained. From the set of samples which pass above test Lmin samples are selected incase if number of samples is greater than $L_{\min}$.
- For *Xi* of first class find the nearest samples of second class retained in step 4.
- Find the point P_i where the straight line connecting the pair of samples found in step 5 meets the decision boundary.
- Find the unit normal vector denoted as *Ni* to the decision boundary at point *Pi* found in step 6.

  $$N = (\sum_1{}^{-1} - \sum_2{}^{-1})^t X_0 + (\sum_1{}^{-1} M_1 - \sum_2{}^{-1} M_2) \quad (3)$$
- The steps 3 to 5 for *Xi*=1, 2, 3….*n*1, *n*1 unit normal vectors will be calculated. Based on the normal vectors calculate the estimate of the effective decision boundary feature matrix ($\sum^1{}_{EDBFM}$) from first class as follows:

  $$\sum{}^1{}_{EDBFM} = 1 / n1 \sum_i^{n1} Ni * Ni^t \quad (4)$$

  The steps 3 to 8 are repeated for second class.
- An estimate of the final effective decision boundary feature matrix is calculated as follows

  $$\sum EDBFM = 1/2 \left(\overset{\square}{\sum} EDBFM + \overset{\square}{\sum} EDBFM \right) \quad (5)$$

  The eigen values help in deciding as to which feature is essentially important for classification. In the above steps only two clusters are used to calculate EDBFM for the sake of simplicity. It resembles a thresholding effect of spilling the entire multispectral imagery to two clusters. Choosing two clusters in the algorithm doesn't harm the calculation of EDBFM to a larger extent. However it can be very well extended to n number of clusters as needed. The main criteria of choosing two clusters are to maintain the simplicity of the pre-process.

3. Customized PSO

The customized PSO is an altered form of PSO for the sake of clustering of multispectral imagery finds the global optima always. The customized PSO which is a modified version of particle swarm model can be very well used as an optimizer. It resembles the behavior of the bird flocking, as PSO does. The particle structure encompasses the cluster centers of the *N* images which is obtained after the reduction of the multispectral imagery using the decision boundary based feature fusion.

3.1. Particle Structure

Each particle structure which resembles the behavior of the bird is considered as the solution. The particles move towards the global optima based on the gbest and pbest. Each of the particle in customized PSO is of length *N*K*, where *N* stands for the number of images obtained after reduction by DBFE and *K* stands for the number of cluster centers required (which is made as the input).In the particles, the values from the position 1 to *K* represents the cluster centers of first image and the values from the position *K+1* to *2*K* represents the cluster centers of the second image and so on. In general the values from position *(N-1)*K+1* to *N*K* represents the cluster centers of the N^{th} image. The particle structure adapted here is well suited for the multispectral image classification. Here the number of particles denoted as pop-size is maintained as a constant. The steps involved in customized PSO are discussed below.

3.2. Initialization

The particles in the customized PSO are initialized using the modified k-Means algorithm. The *k*-Means algorithm which is a well known statistical algorithm provide very good seeds for initialization. The *k* cluster centers are obtained for entire *N* reduced images and used for initialization. The Modified *k*-Means is run for 10 iterations and the values are fed as input for the particle initialization. It is repeated for pop-size times and the entire swarm is initialized.

3.3. Fitness Computation

The fitness is computed for each particle using equation 7. The objective of customized PSO is to reduce the sum of TWCV of the *k* clusters of the entire *N* multispectral imagery, which is defined as sum-of-deviation. Thus, it is a minimization problem. In order to convert the minimization problem to maximization problem a negative sign is included in the exponential part of fitness function. The fitness function reveals a very high value to the particle with very low sum-of-deviation. Based on the fitness value computed the particle with the highest fitness value is assigned as the global best denoted as g-best. In the equation the fitness of the i^{th} particle denoted as P_i is calculated by using an exponential function since it returns 1 (maximum value) if t is zero and the value decreases as t increases. The fitness used here returns a value from 0-1 based on sum-of-deviation of the particle. The sum-of-deviation is found using the below equation and then scaled by multiplying with 1/ (NUM_IMG*m*n) where m and n represents the size of each image in the multispectral imagery.

$$t = sum\text{-}of\text{-}deviation / (NUM\_IMG * m * n) \quad (6)$$

$$Fit(P_i) = exp^{(-.005*t)} \quad (7)$$

$$Sum\text{-}of\text{-}deviation = \sum_{i=1}^{K} TWCV_i \quad (8)$$

3.4. Velocity Updation

The velocity function makes the particles to move towards the global optima based on the value of the particle of the *g*-best. In the customized PSO the velocity function is tailored for the clustering of multispectral images. After each iteration the velocity is updated as a factor of the g-best alone. The velocity function used here does not remember the previous value of earlier iteration. In customized PSO the velocity of the particle P_i is computed using equation 9. The constant *c1* involved in this equation is a random number generated in the interval 0 to 1. The velocity may be positive or negative depending upon how much the particle deviates from *g*-best. In the velocity function the particle best named p-best is not involved, since *g*-best is the best of *p*-best and its enough for the velocity to update based on the g-best.

$$Velocity\ (P_i) = c1 * (g\text{-}best - P_i) \quad (9)$$

3.5. Particle Updation

The particles are updated based on the velocity computed for that particle. The updated particle is found by computing the sum of the particle and the g-best. The sum is done by adding the value of the particle and the velocity in each position. As the particles are updated by velocity, it is dragged towards the global optima. And as generations pass the velocity which updates the particle based on *g*-best makes the entire swarm to converge to global optima.

$$P_i = P_i + velocity\ (P_i) \quad (10)$$

The above steps are repeated until the termination condition is reached. The termination condition can be either any fixed number of generations or until there is no change in the value of the particles between generations. The termination condition, which we have chosen here, is fixed number of generations. It was found customized PSO converges to global optima within 100 generations with 98.56% accuracy, which is higher than the accuracy provided, by the k-Means algorithm.

4. Experimental Study

The customized PSO is demonstrated here using a 3*3 matrix which is obtained by resizing the entire image. The algorithm will run on the entire multispectral imagery and we have chosen the number of images in the imagery set as 3. The multispectral imagery will have much redundant information and before the imagery enters the customized PSO it is given to DBFE and reduced. The 3*3 taken as input is shown below:

$$\begin{bmatrix} 70 & 61 & 66 \\ 76 & 49 & 26 \\ 55 & 78 & 89 \end{bmatrix} \quad \begin{bmatrix} 72 & 62 & 65 \\ 77 & 49 & 27 \\ 54 & 78 & 88 \end{bmatrix} \quad \begin{bmatrix} 71 & 61 & 66 \\ 77 & 50 & 27 \\ 55 & 77 & 89 \end{bmatrix}$$

4.1. Decision Boundary Feature Extraction

The DBFE algorithm [5] is ran over the above matrix of size 3*3 and the steps involved are explained below:

- Here the 3 matrixes are fed as an input and the number of clusters is initialized as two for the sake of simplicity.

- Over the entire 3 images modified *k*-means algorithm is applied for 10-20 iterations and two clusters are found for entire image in the imagery. The modified *k*-means is applied by taking each data point as a vector and clusters are found. The cluster centres found are 50 and 70.
- The mean and covariance of first cluster is 43.33, 234.3333 respectively and the mean and covariance of second cluster are 73.33, 98.2667, respectively. The chi-squared test is applied and outliers are detected based on equation 1. Here R_{t1} is set such that 95% of data passes the chi-squared test. In the following steps only those data vectors of the multispectral imagery which passes chi-squared test are used. The pixels which have passed the chi-squared test in the experiment are 55, 54, 55, 49, 49, and 50. Here 55, 54, 55 and 55, 49, 49 represents the vector of the multispectral imagery.
- Then a chi-squared threshold set is applied using equation 2 to the samples of second class based on mean and covariance of second class. The R_{t2} is set such as 95% of the data set passes the chi-squared test. Those data set which passes equation 2 are retained and used for later steps. The samples which have passed step 4 are 70, 72, 71, 66, 65 and 66. Here 70, 72, 71 and 66, 65, 66 represents the vector in the multispectral imagery. From the set of samples which pass above test lmin samples are selected incase if number of samples is greater than lmin. Here lmin is chosen as 6. Hence all the 6 pixels of the imagery which have passed step 4 are retained as such.
- For the data set of the first class 55, 54, 55, 49, 49, and 50 the nearest samples of second class retained in step 4 are 65, 65, 65, 65, 65, 65 are found and it is preserved.
- The point *Pi* where the straight line connecting the pair of samples are found. For instance the point connecting the pair of samples such as 55 and 65 are found and stored.
- In this experiment the unit normal vector to the decision boundary at point *Pi* is found using equation 3 as 1.0e+003*[0.0002 + 1.6571i, -0.0005 -4.3089i, 0.0002+1.8079i, 0.0002-1.8539i, -0.0005 +4.6862i, 0.0002-1.8840i].
- The decision boundary feature matrix is computed based on normal vectors using equation 4 for first and second class. The EDBFM found in this experiment are shown below here:

$$\sum^{1} EDBFM = 1.0e + 007 \ *$$

$$\left[\begin{array}{lll} 0.6342 & -1.638 + 0.0000i & 0.6552 \\ + 106138 - 0.0000i & 4.1065 & -1.6672 + (\\ 0.6552 + 0.0000i & + 1.6672 + 0.0000i & 0.6769 \end{array}\right.$$

$$\sum^{\square} EDBFM = \square.\square e + \square\square\square \ *$$

$$\left[\begin{array}{lll} \square.\square\square\square\square & -\square.\square\square\square\square + \square.\square\square\square\square i & \square.\square\square\square\square - \square.\square \\ -\square.\square\square\square\square + \square.\square\square\square i & \square.\square\square\square\square & -\square.\square\square\square\square + \square.\square \\ \square.\square\square\square\square + \square.\square\square\square\square i & -\square.\square\square\square\square - \square.\square\square\square\square i & \square.\square\square\square\square \end{array}\right.$$

- Based on equation 5, an estimate of the final effective decision boundary feature matrix is calculated. The EDBFM found in this experiment as follows

$$\sum EDBFM = 1.0e + 007 \ *$$

$$\left[\begin{array}{lll} 1.2525 & -3.1966 + 0.0000i & 1.3041 - 0.000 \\ -3.1966 - 0.0000i & 8.1593 & -3.3291 + 0.000 \\ 1.3041 + 0.0000i & -3.3291 - 0.0000i & 1.3586 \end{array}\right.$$

Based on the eigen value the images with redundant information in the multispectral imagery are found and it is eliminated. The eigen values determined for three images in this experiment are 1.0e+006* [0.0026, 0.6847, 7.5419], respectively. According to the eigen values calculated it is found first image doesn’t contain any new information , hence it can be eliminated for processing in the further steps. The reduced multispectral imagery is fed as an input to the customized PSO.

4.2. Customized PSO

The customized PSO operates on the reduced imagery and finds the global optima of the entire imagery. It was found that customized PSO, which is a tailored format of PSO for the sake of multispectral imagery always converge to the global optima. One of the major problems in working of multispectral imagery is due to the vast dimensionality. The problem is alleviated here by preprocessing due to DBFE. An insight into the steps of customized PSO is shown below.

4.2.1. Particle Structure

A particle structure suitable for the operations of multispectral imagery is adapted in customized PSO. The number of clusters required for the classification of the imagery is made as an input and for this experiment let it be *K*=3. And the number of images obtained after reduction is *N*=2 here. In general the particle is of length *N*K* and hence here the length will be 2*3=6. The k^{th} cluster centre of n^{th} image is obtained by taking the value from the position *n*k* in the particle. One of the particles of this experiment is depicted in Figure 1. From Figure 1, the values 60.4286, 40.6, 75.6 represent cluster centers of first image and 62.5, 33, 77.5 represent cluster centre of the second image.

60.4286	40.6	75.6	62.5	33	77.5

Figure 1. A particle of customized PSO.

4.2.2. Initialization

The seeds for initialization of customized PSO are obtained by running modified *k*-means [8] for 10 iterations. The centers obtained for this experiment by modified *k*-means is shown below:

$$\text{Center:}\begin{bmatrix} 60.4286 & 62.5000 \\ 40.6000 & 33.0000 \\ 75.6000 & 77.5000 \end{bmatrix}$$

The first column represents the centers of first image and the second column represents the cluster centre of second image. These values are used to initialize the particles. The customized PSO operates on these particles from modified *k*-means based initialization. The modified *k*-means is ran pop-size times and the entire swarm is initialized.

4.2.3. Fitness Computation

The fitness used in customized reveals high value for the cluster with lower TWCV. The fitness is obtained using equation 7. The fitness of the particle obtained after eachiteration is tabulated in Table 1. From the tabulated value, it was found as the generation passes the fitness value increases. At the initial stages the fitness is low, but as the customized PSO is applied over the particles the fitness found to increase. The customized PSO is ran for 100 generation and after 100 generation it was found that the fitness of g-best as 98.56%.

4.2.4. Velocity Updation

The velocity of each particle is computed using equation 9. The velocity drags the particles towards global-optima. The velocity of each particle is calculated at the end of eachiteration and it reveals the value how much it deviates from g-best. The g-best is the particle with highest fitness that is found until the end of previous iteration. The velocity may be positive or negative. The velocity of the particle in Figure 2 computed based on the global best is [+1.6, -0.6, +2.4, +1.5, -6, -0.5].

62	40	78	61	27	77

(a) Global best particle.

60.4	40.6	75.6	62.5	33	77.5

(b) A particle in customized PSO.

Figure 2. Particle considered for velocity calculation.

4.2.5. Particle Updation

The particle updation drags the particle towards the g-best (particle with best fitness until previous generation) thereby moves the entire swarm towards the global optima. The particles are updated after eachiteration using equation 10. The particles are updated by summing up the velocity with the particle. The sum is done by adding the velocity and particle vector at each position respectively. As a result of particle updation the particles are moved towards global-optima.

The above steps are repeated for 100 generations and it was found customized PSO always converge to global-optima. The customized PSO converges with 98.56% accuracy. The cluster centers after each *10th* generation is shown in Table 1 and at the end of *100th* generation for the experimental input the cluster centre found is 60.3333, 78.7500, 38.0000, 55.3333, 76.0000, 27.0000.

Table 1. value of g-best after each 10th generation.

Generation	G-BEST					
10	88.50	69.75	51.32	76.56	58.11	27.24
20	64.22	53.16	78.70	60.08	51.03	77.26
30	51.50	78.77	67.51	75.81	27.46	38.54
40	65.89	51.08	78.22	70.87	54.06	90.12
50	67.51	50.72	78.69	60.38	76.57	50.62
60	67.49	50.59	83.11	55.00	71.60	89.44
70	49.61	62.86	81.09	71.70	54.02	90.47
80	67.39	50.85	81.09	71.70	54.02	90.47
90	79.48	66.56	52.00	77.91	28.32	58.08
100	79.48	66.56	52.00	77.91	28.32	58.08

5. Results and Discussion

The customized PSO works well in classification of multispectral imagery and proves to converge to the global optima always. The accuracy of the proposed method for classification of multispectral images is higher (98.56%) than the existing methods [7]. The DBFE is included prior to customize PSO to reduce the dimensionality. Thereby one of the major problems of multispectral imagery due to increase in dimensionality is alleviated. A plot of the time consumed in minutes to find the global optima in a multispectral imagery with and without preprocessing using DBFE is shown in Figure 3, and it very well shows the effectiveness of the DBFE, i.e., the time for obtaining global optimized cluster centers is reduced when DBFE is used prior to clustering compared to the case when DBFE is not used. The customized PSO is tailored appropriately such that it consumes less time to find the global optima in multispectral imagery. A generalized form of implementation for any number of images in a multispectral imagery with required number of clusters (made as input) is done here using matlab and it was found that the customized PSO converges to the global optima always with 98.56% of accuracy. Three images were taken as input is shown in Figure 4-a and the process of feature extraction reduces the size of imagery two. And the classified imagery by customized PSO is shown in Figure 4-b.

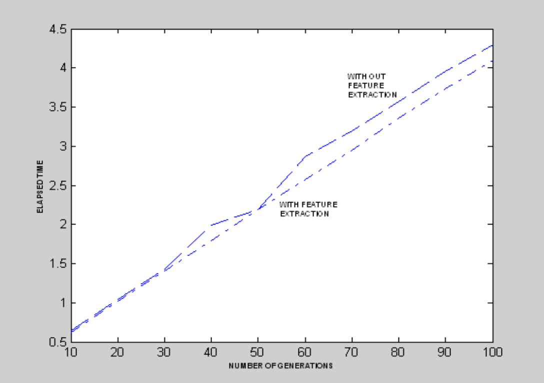
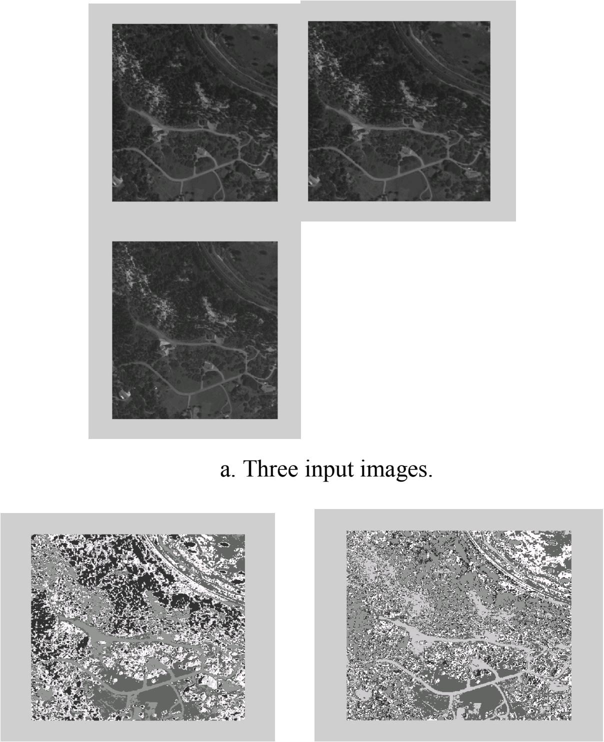


Figure 3. Plot of time elapsed in minutes for customized PSO with and without feature extraction.



a. Three input images.

b. Two output images.

Figure 4. Output images.

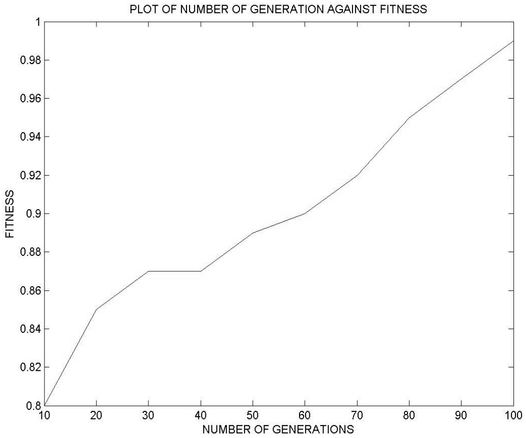


Figure 5. Plot of fitness against number of generations.

6. Conclusion

The customized PSO, modified format of PSO for the sake of classification of multispectral imagery works well and always find the global optima. The reduction in dimensionality using DBFE helps customized PSO to reduce the time consumption. The customized PSO is simplified by modifying velocity update using gbest alone and the value of the velocity of previous generation is not kept in memory. But however the fitness function used to find the best particles can be still simplified and it is considered as a future work. As the number of generation elapses, the convergence of the particle towards global optima leading to the increase in fitness is shown in Figure 5.

References

[1] Babu G. and Murty M., "A Near-Optimal Initial Seed Selection in *K*-Means Algorithm Using a Genetic Algorithm," *Pattern Recognition Letters*, vol. 14, pp. 763-769, 1993.

[2] Fogel D., "An Introduction to Simulated Evolutionary Optimization," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 3-14, 1994.

[3] Jones D. and Beltramo M., "Solving Partitioning Problems with Genetic Algorithms," *in Proceedings of the 4th International Conference Genetic Algorithms*, San Mateo, 1991.

[4] Klein R. and Dubes R., "Experiments in Projection and Clustering by Simulated Annealing," *Pattern Recognition*, vol. 22, pp. 213-220, 1989.

[5] Lee C. and Landgrebe D., "Decision Noundary Feature Selection for Nonparametric Classifiers," *in Proceedings of the SPSES 44 Annual Conference*, pp. 475-478, 1991.

[6] Lee C. and Landgrebe D., "Feature Extraction Based on Decision Boundaries," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 388-400, 1993.

[7] Venkatalakshmi K. and Shalinie M., "Multispectral Image Classification Using Modified k-Means Algorithm," Selected for publication in *Neural Network World*.

[8] Venkatalakshmi K., Sridhar S., and Shalinie M., "Neuro-Statistical Classification of Multispectral Images Based on Decision Fusion," *Neural Network World*, vol. 16, no. 2, pp. 97-107, 2006.

Venkatalakshmi Krishnan is working as a senior lecturer in the Department of Information Technology, Thiagarajar College of Engineering, Madurai, India. She has published papers in national and international conferences and international journals. Her research areas include pattern recognition, evolutionary algorithms, and multispectral data fusion.

Anisha Praisy is doing final year B.Tech (IT) in Thiagarajar College of Engineering, India. She has published papers in national conference and international journals. Her areas of interest include image processing, and pattern recognition.

Maragathavalli R. is doing final year B.Tech (IT) in Thiagarajar College of Engineering, India. She has published papers in national conference and international journals. Her areas of interest include image processing, and evolutionary algorithms.

Mercy Shalinie is working as an assistant professor in the Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India. She has obtained her doctorate from Madurai Kamaraj University in the area of neuro-fuzzy systems for pattern recognition. She has to her credit, 30 papers in National, International Conferences and Journals.