Chapter 4

A PSO-based Clustering Algorithm with Application to Unsupervised Image Classification

A clustering method that is based on PSO is developed in this chapter. The algorithm finds the centroids of a user specified number of clusters, where each cluster groups together similar patterns. The application of the proposed clustering algorithm to the problem of unsupervised classification and segmentation of images is investigated. To illustrate its wide applicability, the proposed algorithm is then applied to synthetic, MRI and satellite images. Experimental results show that the PSO clustering algorithm performs better than *state-of-the-art* clustering algorithms (namely, K-means, Fuzzy C-means, K-Harmonic means and Genetic Algorithms) in all measured criteria. The influence of different values of PSO control parameters on performance is illustrated. The performance of different versions of PSO is also investigated.

4.1 PSO-Based Clustering Algorithm

This section defines the terminology used throughout the rest of the chapter. A measure is given to quantify the quality of a clustering algorithm, after which the PSO-based clustering algorithm is introduced.

4.1.1 Measure of Quality

Different measures can be used to express the quality of a clustering algorithm. The most general measure of performance is the quantization error, defined as

$$J_e = \frac{\sum_{k=1}^{K}\left[\sum_{\forall z_p \in \boldsymbol{C}_k} d(\boldsymbol{z}_p, \boldsymbol{m}_k)\right] \Big/ n_k}{K} \tag{4.1}$$

where $\boldsymbol{C}_k$ is the k^{th} cluster, and n_k is the number of pixels in $\boldsymbol{C}_k$

4.1.2 PSO-Based Clustering Algorithm

In the context of data clustering, a single particle represents the K cluster centroids. That is, each particle $\boldsymbol{x}_i$ is constructed as $\boldsymbol{x}_i = (\boldsymbol{m}_{i,1},\ldots,\boldsymbol{m}_{i,k},\ldots,\boldsymbol{m}_{i,K})$ where $\boldsymbol{m}_{i,k}$ refers to the k^{th} cluster centroid vector of the i^{th} particle. Therefore, a swarm represents a number of candidate data clusterings. The quality of each particle is measured using

$$f(\boldsymbol{x}_i, \boldsymbol{Z}_i) = w_1 \bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) + w_2 (z_{\max} - d_{\min}(\boldsymbol{x}_i)) \tag{4.2}$$

where $z_{\max}$ is the maximum value in the data set (i.e. in the context of digital images, $z_{\max} = 2^s - 1$ for an s-bit image); $\boldsymbol{Z}_i$ is a matrix representing the assignment of patterns to the clusters of particle i. Each element $z_{i,k,p}$ indicates if pattern $\boldsymbol{z}_p$ belongs to cluster $\boldsymbol{C}_k$ of particle i. The constants w_1 and w_2 are user-defined constants used to weigh the contribution of each of the sub-objectives. Also,

$$\bar{d}_{max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) = \max_{k=1,\ldots,K}\left\{\sum_{\forall z_p \in \boldsymbol{C}_{i,k}} d(\boldsymbol{z}_p, \boldsymbol{m}_{i,k}) / n_{i,k}\right\} \tag{4.3}$$

is the maximum average Euclidean distance of particles to their associated clusters, and

$$d_{min}(\boldsymbol{x}_i) = \min_{\forall k,kk,k\neq kk} \{d(\boldsymbol{m}_{i,k}, \boldsymbol{m}_{i,kk})\} \quad (4.4)$$

is the minimum Euclidean distance between any pair of clusters. In the above, $n_{i,k}$ is the number of patterns that belong to cluster $\boldsymbol{C}_{i,k}$ of particle *i*.

The fitness function in equation (4.2) has as objective to simultaneously minimize the intra-distance between patterns and their cluster centroids, as quantified by $\bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i)$, and to maximize the inter-distance between any pair of clusters, as quantified by, $d_{\min}(\boldsymbol{x}_i)$.

According to the definition of the fitness function, a small value of $f(\boldsymbol{x}_i, \boldsymbol{Z}_i)$ suggests compact and well-separated clusters (i.e. *good* clustering).

The fitness function is thus a multi-objective problem. Approaches to solve multi-objective problems have been developed mostly for evolutionary computation approaches [Coello Coello 1996]. Recently, approaches to multi-objective optimization using PSO have been developed by Hu and Eberhart [Multiobjective 2002], Fieldsend and Singh [2002] and Coello Coello and Lechuga [2002]. Since our scope is to illustrate the applicability of PSO to data clustering, and not on multi-objective optimization, a simple weighted approach is used to cope with multiple objectives. Different priorities are assigned to the subobjectives via appropriate initialization of the values of w_1 and w_2.

The PSO clustering algorithm is summarized in Figure 4.1.

1. Initialize each particle to contain K randomly selected cluster centroids

2. For $t = 1$ to t_{max}

(a) For each particle i

i. For each pattern z_p

- calculate $d(\boldsymbol{z}_p, \boldsymbol{m}_{i,k})$ for all clusters $\boldsymbol{C}_{i,k}$ using equation (3.1)
- assign $\boldsymbol{z}_p$ to $\boldsymbol{C}_{i,k}$ where

$$d(\boldsymbol{z}_p, \boldsymbol{m}_{i,k}) = \min_{\forall k=1,\ldots,K} \{d(\boldsymbol{z}_p, \boldsymbol{m}_{i,k})\} \quad (4.5)$$

ii. Calculate the fitness, $f(\boldsymbol{x}_i, \boldsymbol{Z}_i)$

(b) Find the personal best position for each particle and the global best solution, $\hat{\boldsymbol{y}}(t)$

(c) Update the cluster centroids using equations (2.8) and (2.10)

Figure 4.1: The PSO clustering algorithm

As previously mentioned, an advantage of using PSO is that a parallel search for an optimal clustering is performed. This population-based search approach reduces the effect of the initial conditions, compared to K-means (as shown in Figure 4.4), especially for relatively large swarm sizes.

4.1.3 A Fast Implementation

Since most of the images used in this thesis are single band, gray scale images and since most clustering algorithms do not use spatial information, a fast implementation

is used for this type of images in order to speedup the execution time of the algorithms used. The fast implementation works as follows:

1) The histogram of a single band, gray scale image is created by calculating the frequency of each gray level.
2) A data structure is used where each gray level is associated with a frequency value and a cluster label.
3) Depending on the algorithm used, perform all the calculations (e.g. Euclidean distance, calculation of centroids, fitness function, etc.) using the above data structure by multiplying each gray level by its frequency and using the cluster labels for clustering.

Using the above implementation, the execution time will be independent on the size of the image. However, the execution time will depend on the number of gray levels which is usually very small (e.g. 256 for 8-bit images and 1024 for 10-bit images). Furthermore, the number of gray levels is generally much less than the number of pixels. Hence, the execution time will reduce significantly while the results are the same. Therefore, this implementation is used in this thesis for single band, gray scale images.

4.2 Experimental Results

The PSO-based clustering algorithm has been applied to three types of imagery data, namely synthetic, MRI and LANDSAT 5 MSS (79 m GSD) images. These data sets have been selected to test the algorithms, and to compare them with other algorithms, on a range of problem types, as listed below:

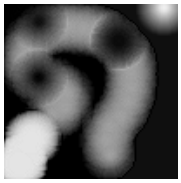
Synthetic Image: Figure 4.2(a) shows a 100 × 100 8-bit gray scale image created to specifically show that the PSO algorithm does not get trapped in the local minimum. The image was created using two types of brushes, one brighter than the other.

MRI Image: Figure 4.2(b) shows a 300 × 300 8-bit gray scale image of a human brain, intentionally chosen for its importance in medical image processing.

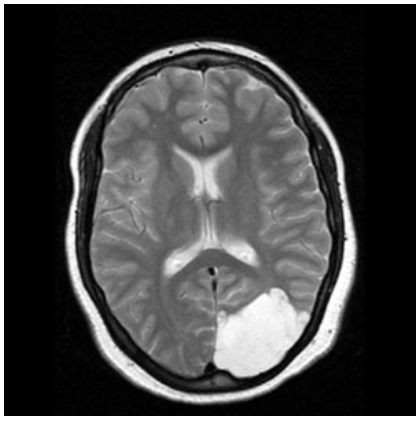
Remotely Sensed Imagery Data: Figure 4.2(c) shows band 4 of the four-channel multispectral test image set of the Lake Tahoe region in the US. Each channel is comprised of a 300 × 300, 8-bit per pixel (remapped from the original 6 bit) image. The test data are one of the North American Landscape Characterization (NALC) Landsat multispectral scanner data sets obtained from the U.S. Geological Survey (USGS).

The rest of this section is organized as follows: Section 4.2.1 illustrates that the basic PSO can be used successfully as an unsupervised image classifier, using the original fitness function as defined in equation (4.2). Section 4.2.2 illustrates the performance under a new fitness function. Results of the *gbest* PSO are compared with that of GCPSO in section 4.2.3, using the new fitness function. Section 4.2.4 investigates the influence of the different PSO control parameters. The performance of PSO using the new fitness function is compared with *state-of-the-art* clustering approaches in Section 4.2.5. In section 4.2.6, the performance of different versions of PSO is investigated. A new non-parametric fitness function is presented in Section 4.2.7. In section 4.2.8, the PSO-based clustering algorithm is applied to multispectral

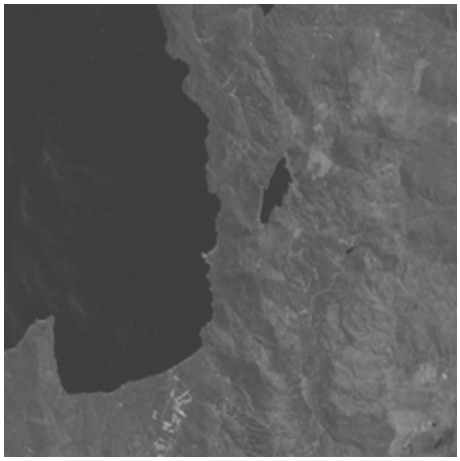
imagery data. Finally, section 4.2.9 provides a discussion of applying PSO to data clustering.



(a) Synthetic image



(b) MRI Image of Human brain



(c) Band 4 of the Landsat MSS test image of Lake Tahoe

Figure 4.2: Data set consisting of synthetic, MRI and LANDSAT images

The results reported in this section are averages and standard deviations over 20 simulations. All comparisons are made with reference to J_e, $\bar{d}_{\max}$ and $d_{\min}$.

Furthermore, a total number of clusters of 3, 8 and 4 were used respectively for the synthetic, MRI and Tahoe images.

4.2.1 *gbest* PSO versus K-Means

This section presents results to compare the performance of the *gbest* PSO algorithm with that of the K-means algorithm for each of the images.

Table 4.1 summarizes the results for the three images. In all cases, for PSO, 50 particles were trained for 100 iterations; for K-means, 5000 iterations were used (that is, both algorithms have performed 5000 function evaluations). $V_{max} = 5$, $w = 0.72$ and $c_1 = c_2 = 1.49$. The chosen values of w, c_1, and c_2 are popular in the literature and ensure convergence [Van den Bergh 2002]. For the fitness function in equation (4.2), $w_1 = w_2 = 0.5$ to give each subobjective an equal contribution.

The results showed that, for the images used, K-means performed better than the PSO algorithm with reference to the quantization error J_e. However, J_e does not give an idea of the quality of the individual clusters. With respect to the minimization of intra-distances ($\bar{d}_{\max}$) and the maximization of inter-distances ($d_{\min}$), the PSO algorithm generally performed better than K-means clustering.

Figure 4.3 illustrates for the synthetic image how the fitness of PSO improves over time. For this figure, 10 particles have been used for a training phase of 100 iterations, $V_{max} = 5$, $w = 0.72$, $c_1 = c_2 = 1.49$, and $w_1 = w_2 = 0.5$. The fitness value, as measured using equation 4.2, improves from the initial 96.637 to 91.781.

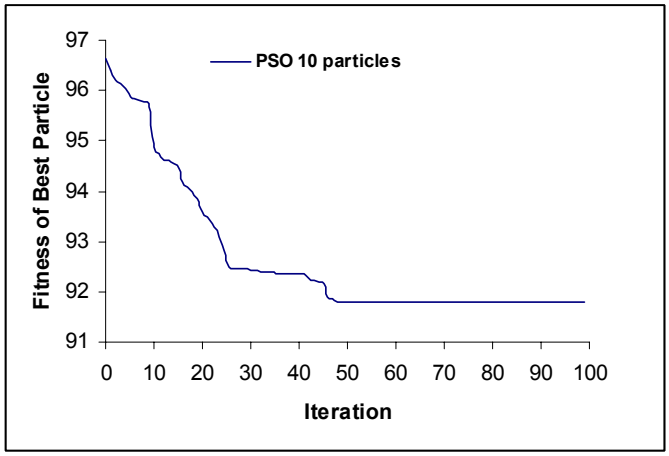


Figure 4.3: PSO Performance on Synthetic Image

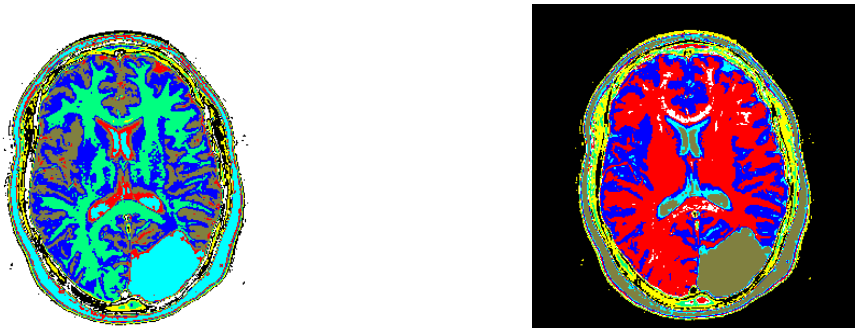
Figure 4.4(a) illustrates the segmented image of the synthetic image for the K-means algorithm, while Figure 4.4(b) illustrates the segmented image obtained from the PSO algorithm. These figures clearly illustrate that K-means was trapped in a local optimum. Three clusters were created using two brushers, the brighter brush were used to create the two spots in the upper right and lower left corner while the other brush were used to create the remaining shape. K-means could not classify the clusters correctly, since it failed to cluster the two spots as separate clusters. PSO, on the other hand, was not trapped in this local optimum and succeeded in showing the two spots as separate clusters. The segmented images for the MRI and the Tahoe images are given in Figures 4.5 and 4.6, respectively.

Table 4.1: Comparison between K-means and PSO

Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	**K-means**	20.212 ± 0.938	28.040 ± 2.778	78.498 ± 7.0629
	PSO	24.453 ± 0.209	27.157 ± 0.017	98.679 ± 0.023
MRI	**K-means**	7.370 ± 0.0428	13.214 ± 0.762	9.934 ± 7.309
	PSO	8.536 ± 0.584	10.129 ± 1.262	28.745 ± 2.949
Tahoe	**K-means**	1.664 ± 0.040	3.107 ± 0.168	4.527 ± 1.347
	PSO	7.215 ± 2.393	9.036 ± 3.363	25.777 ± 9.602

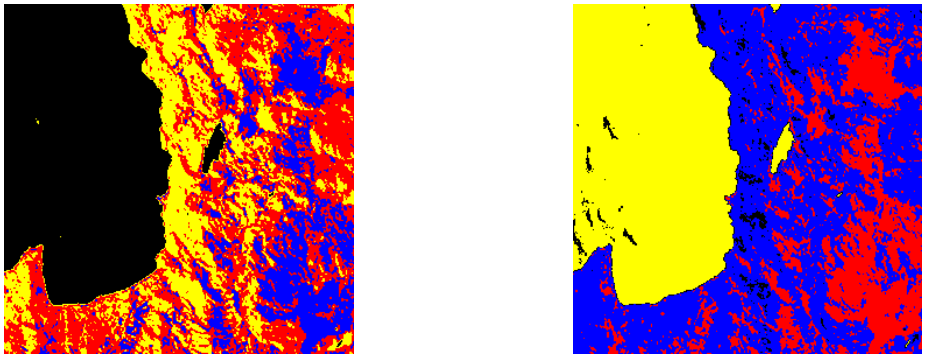
(a) K-means (b) PSO

Figure 4.4: The Segmented Synthetic Images



(a) K-means (b) PSO

Figure 4.5: The Segmented MRI Images



(a) K-means (b) PSO

Figure 4.6: The Segmented Lake Tahoe Images

4.2.2 Improved Fitness Function

The above experimental results have shown that the PSO clustering algorithm improves on the performance of the K-means algorithm in terms of inter- and intra-cluster distances. An improved fitness function which simply adds to the previous fitness function an additional sub-objective to also minimize the quantization error is presented in the following equation:

$$f(\boldsymbol{x}_i, \boldsymbol{Z}_i) = w_1 \bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) + w_2(z_{\max} - d_{\min}(\boldsymbol{x}_i)) + w_3 J_{e,i} \tag{4.6}$$

In this section, the results of the *gbest* PSO shown in the previous section are compared with results using the new fitness function as defined in equation (4.6). All parameters are set as in the previous section. The only difference is that for the extended fitness function, $w_1 = w_2 = 0.3$, $w_3 = 0.4$ were used for the synthetic image, $w_1 = 0.2$, $w_2 = 0.5$, $w_3 = 0.3$ were used for the MRI image and $w_1 = w_2 = w_3 = 0.333333$ were used for the Tahoe image. These values were set empirically.

Table 4.2 compares the results for the two fitness functions. The new fitness function succeeded in significant improvements in the quantization error, J_e. The new fitness function also achieved significant improvements in minimizing the intra-cluster distances for the synthetic and Tahoe images, thus resulting in more compact clusters, and only marginally worse for the MRI image. These improvements were at the cost of loosing on maximization of the inter-cluster distances. However, this loss is acceptable because the *gbest* PSO using the new fitness function still performs

better than the K-means algorithm in terms of the inter-cluster distance (compare the results in Table 4.1 and Table 4.2).

Due to the improved performance on the quantization error and intra-cluster distances, the rest of this chapter uses the 3-component fitness function as defined in equation (4.6).

Table 4.2: 2-component versus 3-component fitness function

Problem	2-Component Fitness Function			3-Component Fitness Function		
	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	24.453 ± 0.209	27.157 ± 0.017	98.679 ± 0.023	17.113 ± 0.548	24.781 ± 0.270	92.768 ± 4.043
MRI	8.536 ± 0.584	10.129 ± 1.262	28.745 ± 2.949	7.225 ± 0.552	12.206 ± 2.507	22.936 ± 8.311
Tahoe	7.215 ± 2.393	9.036 ± 3.363	25.777 ± 9.602	3.556 ± 0.140	4.688 ± 0.260	14.987 ± 0.425

4.2.3 *gbest* PSO versus GCPSO

This section compares the performance of the *gbest* PSO with the GCPSO. This is done for a low $V_{max} = 5$ and a high $V_{max} = 255$. All other parameters are as for section 4.2.2. Table 4.3 shows no significant difference in the performance between PSO and GCPSO. It is, however, important to note that too much clamping of the velocity updates have generally a negative influence on performance. In general, better results were obtained, for both the PSO and GCPSO with a large value of V_{max}.

Table 4.3: PSO versus GCPSO						
Problem	**PSO**			**GCPSO**		
V_{max}=5	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	17.112672 ± 0.548096	24.781384 ± 0.270409	92.767925 ± 4.043086	17.116036 ± 0.547317	24.826868 ± 0.237154	92.845323 ± 4.056681
MRI	7.225384 ± 0.552381	12.205947 ± 2.506827	22.935786 ± 8.310654	7.239264 ± 0.475250	12.438016 ± 2.437064	23.377287 ± 6.722787
Tahoe	3.556281 ± 0.139881	4.688270 ± 0.259919	14.986923 ± 0.425077	3.542732 ± 0.109415	4.672483 ± 0.129913	15.007491 ± 0.621020
V_{max}=255						
Synthetic	17.004993 ± 0.086698	24.615665 ± 0.143658	93.478081± 0.276109	17.000393 ± 0.022893	24.672107 ± 0.174457	93.588530 ± 0.400137
MRI	7.640622 ± 0.514184	10.621452 ± 1.284735	24.948486 ± 3.446673	7.694498 ± 0.591383	10.543233 ± 1.038114	25.355967 ± 3.945929
Tahoe	3.523967 ± 0.172424	4.681492 ± 0.110739	14.664859 ± 1.177861	3.609807 ± 0.188862	4.757948 ± 0.227090	15.282949 ± 1.018218

4.2.4 Influence of PSO Parameters

The PSO have a number of parameters that have an influence on the performance of the algorithm. These parameters include V_{max}, the number of particles, the inertia weight and the acceleration constants. Additionally, the PSO-based clustering algorithm adds a weight to each sub-objective. This section investigates the influence of different values of these parameters.

Velocity Clamping

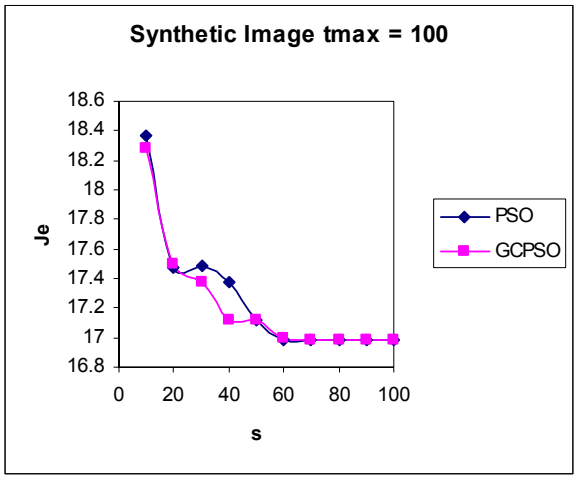
Table 4.3 shows that less clamping of velocity updates is more beneficial. This allows particles to make larger jumps in the search space.

Swarm Size

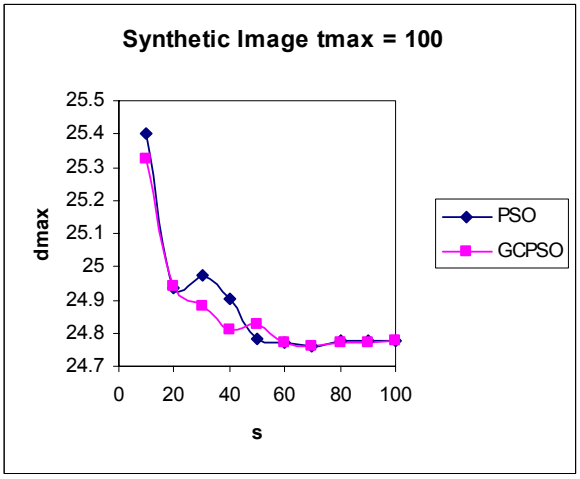
To investigate the effect of different swarm sizes on performance, both the PSO and GCPSO have been executed using 10 to 100 particles. All other parameters are as for section 4.2.2. Figure 4.7 shows the effect of the swarm size, s, on the synthetic image. It is clear from the figure that increasing the number of particles improves the performance of both algorithms. The same conclusion can be drawn for the MRI image as illustrated in Figure 4.8. However, it can be observed from Figure 4.7, that no significant improvement is achived for more than 60 particles. In general, an increase in the number of particles increases diversity, thereby limiting the effects of initial conditions and reducing the possibility of being trapped in local minima.

Inertia Weight

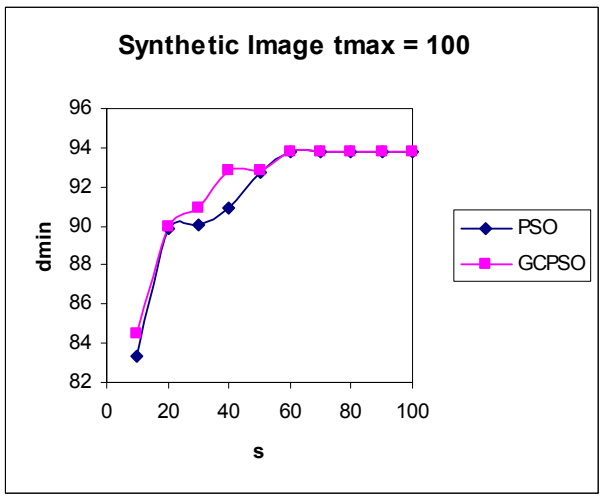
Given that all parameters are fixed at the values given in section 4.2.2, the inertia weight w was set to different values for both PSO and GCPSO. In addition, a dynamic inertia weight was used with an initial w =1.4, which linearly decreased to 0.8. The initial large value of w favors exploration in the early stages, with more exploitation in the later stages with the smaller values. Tables 4.4 and 4.5 summarize the results for the synthetic and MRI images respectively. For the synthetic image, the results illustrate no significant difference in performance, meaning that for the synthetic image, the PSO-based clustering algorithms are generally insensitive to the value of the inertia weight (provided that c_1 and c_2 are selected such that equation (2.9) is not violated). However, in the MRI image, it can be observed that w =0 yields the best results in terms of inter- and intra-cluster distances.



(a) Quantization error

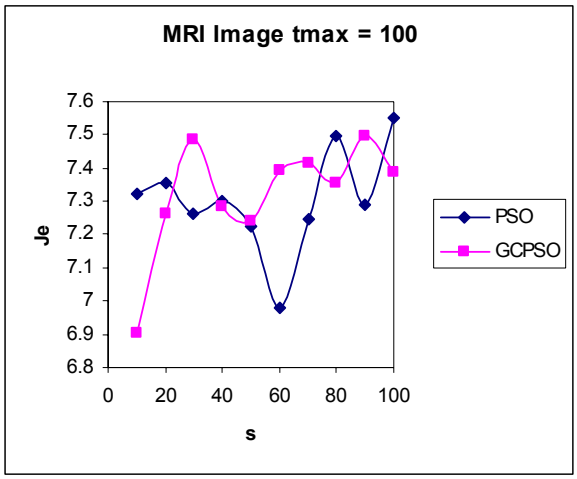


(b) Intra-cluster Distances

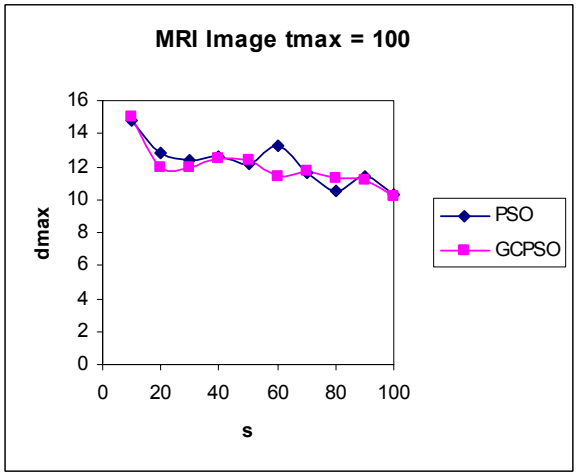


(c) Inter-cluster Distances

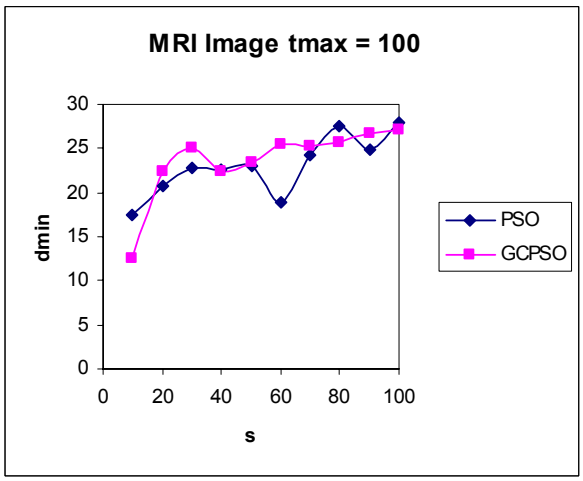
Figure 4.7: Effect of swarm size on synthetic image



(a) Quantization error



(b) Intra-cluster Distances



(c) Inter-cluster Distances

Figure 4.8: Effect of swarm size on MRI image

Table 4.4: Effect of inertia weight on the synthetic image						
w	**PSO**			**GCPSO**		
	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
0.0	16.983429 ± 0.017011	24.581799 ± 0.165103	93.435221 ± 0.308601	16.986386 ± 0.016265	24.649368 ± 0.138223	93.559275 ± 0.254670
0.1	16.982362 ± 0.016074	24.645884 ± 0.137442	93.543795 ± 0.256700	16.985079 ± 0.016995	24.637893 ± 0.138894	93.538635 ± 0.257167
0.5	16.985826 ± 0.014711	24.664421 ± 0.144252	93.595394 ± 0.246110	16.987470 ± 0.028402	24.662973 ± 0.163768	93.58124 ± 0.281366
0.72	16.992102 ± 0.021756	24.670338 ± 0.150542	93.606400 ± 0.258548	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954
0.9	16.993759 ± 0.014680	24.650337 ± 0.140005	93.569595 ± 0.252781	17.040990 ± 0.168017	24.633802 ± 0.352785	93.495340 ± 0.584424
1.4 to 0.8	17.824495 ± 0.594291	24.433770 ± 1.558219	92.625088 ± 2.031224	17.481146 ± 0.504740	24.684407 ± 1.010815	93.223498 ± 1.490217

Table 4.5: Effect of inertia weight on the MRI image						
w	**PSO**			**GCPSO**		
	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
0.0	7.538669 ± 0.312044	9.824915 ± 0.696940	28.212823 ± 2.300930	7.497944 ± 0.262656	9.731746 ± 0.608752	28.365827 ± 1.882164
0.1	7.511522 ± 0.281967	10.307791 ± 1.624499	27.150801 ± 3.227550	7.309289 ± 0.452103	10.228958 ± 1.354945	26.362349 ± 3.238452
0.5	7.612079 ± 0.524669	10.515242 ± 1.103493	26.996556 ± 2.161969	7.466388 ± 0.492750	10.348044 ± 1.454050	26.790056 ± 2.830860
0.72	7.57445 ± 0.382172	10.150214 ± 1.123441	27.393498 ± 3.260418	7.467591 ± 0.396310	10.184191 ± 0.955129	26.596493 ± 3.208689
0.9	7.847689 ± 0.529134	10.779765 ± 1.134843	26.268057 ± 3.595596	7.598518 ± 0.516938	10.916945 ± 1.534848	25.417859 ± 3.174232
1.4 to 0.8	8.354957 ± 0.686190	13.593536 ± 2.035889	21.625623 ± 4.507230	8.168068 ± 0.709875	12.722139 ± 1.850957	21.169304 ± 4.732452

Acceleration Coefficients

Given that all parameters are fixed at the values given in section 4.2.2, the influence of different values for the acceleration coefficients, c_1 and c_2, were evaluated for the synthetic and MRI images. Tables 4.6 and 4.7 summarize these results. For these choices of the acceleration coefficients, no single choice is superior to the others. While these tables indicate an independence to the value of the acceleration coefficients, it is important to note that convergence depends on the relationship between the inertia weight and the acceleration coefficients, as derived in Van den Bergh [2002] (also refer to equation (2.9)).

Table 4.6: Effect of acceleration coefficients on the synthetic image

W	PSO			GCPSO		
	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
$c_1 = 0.7$ $c_2 = 1.4$	16.989197 ± 0.011786	24.726716 ± 0.101239	93.698591 ± 0.184244	16.989355 ± 0.012473	24.708151 ± 0.120168	93.667144 ± 0.207355
$c_1 = 1.4$ $c_2 = 0.7$	16.991884 ± 0.016970	24.700627 ± 0.125603	93.658673 ± 0.208500	16.993095 ± 0.040042	24.685461 ± 0.165669	93.619162 ± 0.279258
$c_1 = 1.49$ $c_2 = 1.49$	16.987582 ± 0.009272	24.710933 ± 0.122622	93.672792 ± 0.206395	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954

Table 4.7: Effect of acceleration coefficients on the MRI image

W	PSO			GCPSO		
	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
$c_1 = 0.7$ $c_2 = 1.4$	7.599324 ± 0.289702	10.14501 ± 1.353091	26.977217 ± 3.467738	7.530823 ± 0.477134	10.201762 ± 0.986726	26.425638 ± 3.248949
$c_1 = 1.4$ $c_2 = 0.7$	7.528712 ± 0.439470	10.23899 ± 1.484245	27.747333 ± 2.850575	7.476468 ± 0.459432	10.159019 ± 1.085977	27.001444 ± 3.360799
$c_1 = 1.49$ $c_2 = 1.49$	7.499845 ± 0.416682	10.20391 ± 0.951100	26.629647 ± 2.652593	7.467591 ± 0.396310	10.184191 ± 0.955129	26.596493 ± 3.208689

Sub-objective Weight Values

Tables 4.8 and 4.9 summarize the effects of different values of the weights, w_1, w_2 and w_3, of the sub-objectives for the synthetic and MRI images respectively. The results show that increasing the value of a weight, improves the corresponding fitness term. However, it is not so clear which sub-objective weight value combination is best for the synthetic and MRI images. To eliminate tuning of these weight values, an alternative multi-objective approach can be followed [Coello Coello 1996; Hu and Eberhart, Multiobjective 2002; Fieldsend and Singh 2002; Coello Coello and Lechuga 2002], or a non-parametric fitness function can be used as proposed in section 4.2.7.

4.2.5 *gbest* PSO versus *state-of-the-art* clustering algorithms

This section compares the performance of the *gbest* PSO and GCPSO with K-means, FCM, KHM, H2 and a GA clustering algorithm. This is done for a high V_{max} = 255. All other parameters are as for section 4.2.2. In all cases, for PSO, GCPSO and GA, 50 particles were trained for 100 iterations; for the other algorithms 5000 iterations were used (i.e. all algorithms have performed 5000 function evaluations). For FCM, q was set to 2 since it is the commonly used value [Hoppner *et al*. 1999]. For KHM and H2, α was set to 2.5 and 4 respectively since these values produced the best results according to our preliminary tests. For the GA, a tournament size of 2 was used, a uniform crossover probability of 0.8 with mixing ratio of 0.5, and a mutation probability of 0.05. Only the best individual survived to the next generation. The results are summarized in Table 4.10. These results are also averages over 20 simulation runs. Table 4.10 shows that PSO and GCPSO generally outperformed K-means, FCM, KHM and H2 in $d_{\min}$ and $\bar{d}_{\max}$, while performing comparably with

respect to J_e (for the synthetic image, PSO performs significantly better than K-means, FCM, KHM and H2 with respect to J_e). The PSO, GCPSO and GA showed similar performance, with no significant difference.

These results show that the PSO-based clustering algorithms are viable alternatives that merit further investigation.

Table 4.8: Effect of sub-objective weight values on synthetic image

			PSO			**GCPSO**		
w_1	w_2	w_3	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
0.3	0.3	0.3	17.068816 ± 0.157375	24.67201 ± 0.572276	93.59498 ± 0.984724	17.01028 ± 0.059817	24.74227 ± 0.258118	93.711385 ± 0.437418
0.8	0.1	0.1	17.590421 ± 0.353375	21.76629 ± 0.127098	88.89228 ± 0.143159	17.51434 ± 0.025242	21.72462 ± 0.018983	88.879342 ± 0.062452
0.1	0.8	0.1	18.827495 ± 0.558357	27.62398 ± 0.427120	97.71945 ± 0.202744	18.82712 ± 0.688529	27.52239 ± 0.282601	97.768398 ± 0.266885
0.1	0.1	0.8	16.962755 ± 0.003149	24.49574 ± 0.089611	93.22883 ± 0.135893	16.98372 ± 0.122501	24.54688 ± 0.434417	92.576271 ± 4.357444
0.1	0.45	0.45	17.550448 ± 0.184982	26.70792 ± 0.692239	96.02056 ± 0.757185	17.55782 ± 0.226305	26.59844 ± 0.907974	95.888089 ± 1.152158
0.45	0.45	0.1	18.134349 ± 0.669151	26.48904 ± 0.982256	96.46178 ± 1.495491	18.29490 ± 0.525467	26.79529 ± 0.800436	96.922471 ± 1.225336
0.45	0.1	0.45	17.219807 ± 0.110357	22.63196 ± 0.522369	90.15281 ± 0.887423	17.20169 ± 0.093969	22.70116 ± 0.469470	90.289690 ± 0.828522

Table 4.9: Effect of sub-objective weight values on MRI image								
			PSO			**GCPSO**		
w_1	w_2	w_3	J_e	$\bar{d}_{\max}$	$d_{\min}$	J_e	$\bar{d}_{\max}$	$d_{\min}$
0.3	0.3	0.3	7.239181 ± 0.576141	10.235431 ± 1.201349	24.705469 ± 3.364803	7.194243 ± 0.573013	10.403608 ± 1.290794	23.814072 ± 3.748753
0.8	0.1	0.1	7.364818 ± 0.667141	9.683816 ± 0.865521	24.021787 ± 3.136552	7.248268 ± 0.474639	9.327774 ± 0.654454	23.103375 ± 4.970816
0.1	0.8	0.1	8.336001 ± 0.599431	11.256763 ± 1.908606	31.106734 ± 1.009284	8.468620 ± 0.626883	11.430190 ± 1.901736	30.712733 ± 1.336578
0.1	0.1	0.8	6.160486 ± 0.241060	15.282308 ± 2.300023	2.342706 ± 5.062570	6.088302 ± 0.328147	15.571290 ± 2.410393	1.659674 ± 4.381048
0.1	0.45	0.45	7.359711 ± 0.423120	10.826327 ± 1.229358	24.536828 ± 3.934388	7.303304 ± 0.439635	11.602263 ± 1.975870	22.939088 ± 3.614108
0.45	0.45	0.1	8.001817 ± 0.391616	9.885342 ± 0.803478	28.057459 ± 1.947362	7.901145 ± 0.420714	9.657340 ± 0.947210	29.236420 ± 1.741987
0.45	0.1	0.45	6.498429 ± 0.277205	11.392347 ± 2.178743	12.119429 ± 8.274427	6.402205 ± 0.363938	10.939902 ± 2.301587	14.422413 ± 6.916785

Table 4.10: Comparison between K-means, FCM, KHM, H2, GA and PSO for fitness function defined in equation (4.6)

Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	**K-means**	20.21225 ± 0.937836	28.04049 ± 2.7779388	78.4975 ± 7.0628718
	FCM	20.731920 ± 0.650023	28.559214 ± 2.221067	82.434116 ± 4.404686
	KHM(p=2.5)	20.168574 ± 0.0	23.362418 ± 0.0	86.307593 ± 0.000008
	H2 (p=4)	20.136423 ± 0.793973	26.686939 ± 3.011022	81.834143 ± 6.022036
	GA	17.004002 ± 0.035146	24.603018 ± 0.11527	93.492196 ± 0.2567
	PSO	16.988910 ± 0.023937	24.696055 ± 0.130334	93.632200 ± 0.248234
	GCPSO	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954
MRI	**K-means**	7.3703 ± 0.042809	13.214369 ± 0.761599	9.93435 ± 7.308529
	FCM	7.205987 ± 0.166418	10.851742 ± 0.960273	19.517755 ± 2.014138
	KHM(p=2.5)	7.53071± 0.129073	10.655988 ± 0.295526	24.270841 ± 2.04944
	H2 (p=4)	7.264114 ± 0.149919	10.926594 ± 0.737545	20.543530 ± 1.871984
	GA	7.038909 ± 0.508953	9.811888 ± 0.419176	25.954191 ± 2.993480
	PSO	7.594520 ± 0.449454	10.186097 ± 1.237529	26.705917 ± 3.008073
	GCPSO	7.555421 ± 0.409742	9.983189 ± 0.915289	27.313118 ± 3.342264
Tahoe	**K-means**	3.280730 ± 0.095188	5.234911 ± 0.312988	9.402616 ± 2.823284
	FCM	3.164670 ± 0.000004	4.999294 ± 0.000009	10.970607 ± 0.000015
	KHM(p=2.5)	3.830761 ± 0.000001	6.141770 ± 0.0	13.768387 ± 0.000002
	H2 (p=4)	3.197610 ± 0.000003	5.058015 ± 0.000007	11.052893 ± 0.000012
	GA	3.472897 ± 0.151868	4.645980 ± 0.105467	14.446860 ± 0.857770
	PSO	3.523967 ± 0.172424	4.681492 ± 0.110739	14.664859 ± 1.177861
	GCPSO	3.609807 ± 0.188862	4.757948 ± 0.227090	15.282949 ± 1.018218

4.2.6 Different Versions of PSO

This section investigates the use of different versions of PSO, namely:

- *lbest* PSO (with $l = 2$).
- *gbest-to-lbest* PSO, starting start with an *lbest* implementation of the PSO (with zero-radius neighborhood i.e. $l = 0$) and linearly increasing the neighborhood radius until a *gbest* implementation of the PSO is reached. This hybrid approach is used in order to initially explore more, thus, avoid being trapped in local optima, by using a *lbest* approach [Suganthan 1999]. The algorithm then attempts to converge onto the best solution found by the initial phase by using a *gbest* approach.
- *gbest*- and *lbest*- PSO with mutation proposed by Esquivel and Coello Coello [2003] (discussed in section 2.6.8). In this approach, the PSO parameters where set as specified by Esquivel and Coello Coello [2003] (i.e. $w = 0.3$, $c_1 = c_2 = 1.3$). In addition, p_m is initialized to 0.9, then linearly decreases with increase in the number of iterations [Coello Coello 2003].

Table 4.11 summarizes the results of *gbest* PSO, GCPSO, *lbest* PSO, *gbest-to-lbest* PSO, *gbest* PSO with mutation and *lbest* PSO with mutation. In all the experiments, 50 particles were trained for 100 iterations and $V_{max} = 255$. All the other parameters are as for section 4.2.2 for all the approaches except the approaches that use mutation. Although the results are generally comparable, it can be observed that the *gbest-to-lbest* PSO is slightly better than the others. An explanation for this observation is the fact that *gbest-to-lbest* PSO starts with high diversity (therefore more exploration),

then as the run progresses, the diversity is reduced (to focus more exploitation). This observation shows the importance of high diversity at the beginning of the run in order to avoid premature convergence and the importance of low diversity at the end of the run in order to fine tune the solution.

Table 4.11: Comparison of different PSO versions

Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	***gbest* PSO**	16.988910 ± 0.023937	24.696055 ± 0.130334	93.632200 ± 0.248234
	GCPSO	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954
	***lbest* PSO**	16.991791 ± 0.003523	24.771597 ± 0.004171	93.775989 ± 0.0
	***gbest*-to-*lbest* PSO**	16.988325 ± 0.000273	24.774668 ± 0.000371	93.775989 ± 0.0
	***gbest* PSO with mutation**	16.985563 ± 0.006350	24.728548 ± 0.110016	93.698591 ± 0.184244
	***lbest* PSO with mutation**	16.995550 ± 0.015914	24.684511 ± 0.135164	93.646993 ± 0.223429
MRI	**PSO**	7.594520 ± 0.449454	10.186097 ± 1.237529	26.705917 ± 3.008073
	GCPSO	7.555421 ± 0.409742	9.983189 ± 0.915289	27.313118 ± 3.342264
	***lbest* PSO**	7.676197 ± 0.138833	9.500085 ± 0.567423	29.684682 ± 0.929038
	***gbest*-to-*lbest* PSO**	7.663361 ± 0.142196	9.211712 ± 0.502518	30.138389 ± 0.878266
	***gbest* PSO with mutation**	7.301802 ± 0.474767	9.573999 ± 0.581114	27.691924 ± 3.145707
	***lbest* PSO with mutation**	7.657294 ± 0.277544	9.890083 ± 0.696923	28.731981 ± 1.938404
Tahoe	**PSO**	3.523967 ± 0.172424	4.681492 ± 0.110739	14.664859 ± 1.177861
	GCPSO	3.609807 ± 0.188862	4.757948 ± 0.227090	15.282949 ± 1.018218
	***lbest* PSO**	3.527251 ± 0.212840	4.778272 ± 0.217206	15.619541 ± 1.179783
	***gbest*-to-*lbest* PSO**	3.460024 ± 0.289942	4.826269 ± 0.238982	15.985762 ± 1.410871
	***gbest* PSO with mutation**	3.592122± 0.180782	4.750996 ± 0.213625	15.252226 ± 0.987399
	***lbest* PSO with mutation**	3.660723± 0.121711	4.793518 ± 0.144508	15.522304 ± 0.597297

4.2.7 A Non-parametric Fitness Function

The fitness function defined in equation (4.6) provides the user with the flexibility of prioritizing the fitness term of interest by modifying the corresponding weight. However, it requires the user to find the best combination of w_1, w_2 and w_3 for each image which is not an easy task. Therefore, a non-parametric fitness function without weights is defined as

$$f(\boldsymbol{x}_i, \boldsymbol{Z}_i) = \frac{\bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) + J_{e,i}}{d_{\min}(\boldsymbol{Z}_i, \boldsymbol{x}_i)} \tag{4.7}$$

The advantage of equation (4.7) is that it works with any data set without any user intervention. Table 4.12 is a repeat of Table 4.10, but with the results of *gbest* PSO using the non-parametric fitness function (referred to as *PSO noweights*) added. In general, the PSO using the non-parametric fitness function performed better than K-Means, FCM, KHM and H2 in terms of $d_{\min}$ and $\bar{d}_{\max}$, while performing comparably with respect to J_e. In addition, the PSO using the non-parametric fitness function performed comparably with GA, PSO and GCPSO using the parametric fitness function (equation (4.6)). Hence, the non-parametric fitness function (equation (4.7)) can be used instead of the parametric fitness function (equation (4.6)), thereby eliminating the need for tuning w_1, w_2 and w_3.

However, since the difference between J_e and $\bar{d}_{\max}$ on the one hand and $d_{\min}$ on the other hand is quite large (as can be observed from the results of this section), the value of the fitness function is usually less than one, and may incorrectly indicate

a good clustering for large values of J_e and $\bar{d}_{\max}$. The proposed non-parametric fitness function therefore has the problem that the largest criterion tends to dominate the other criteria. To address this biased behavior, the values of J_e and $\bar{d}_{\max}$ are normalized to the range [0,0.5], while the value of $d_{\min}$ are normalized to the range [0,1]. Therefore, J_e + $\bar{d}_{\max}$ and $d_{\min}$ contributes equally the fitness function. Table 4.13 compares the performance of the non-normalized, non-parametric fitness function (*PSO noweights*) with the normalized, non-parametric fitness function (*PSO normalized noweights*). From Table 4.13, it can be observed that both non-parametric fitness functions performed comparably. Hence, it can be concluded that it is not necessary to normalize the non-parametric fitness function.

4.2.8 Multispectral Imagery Data

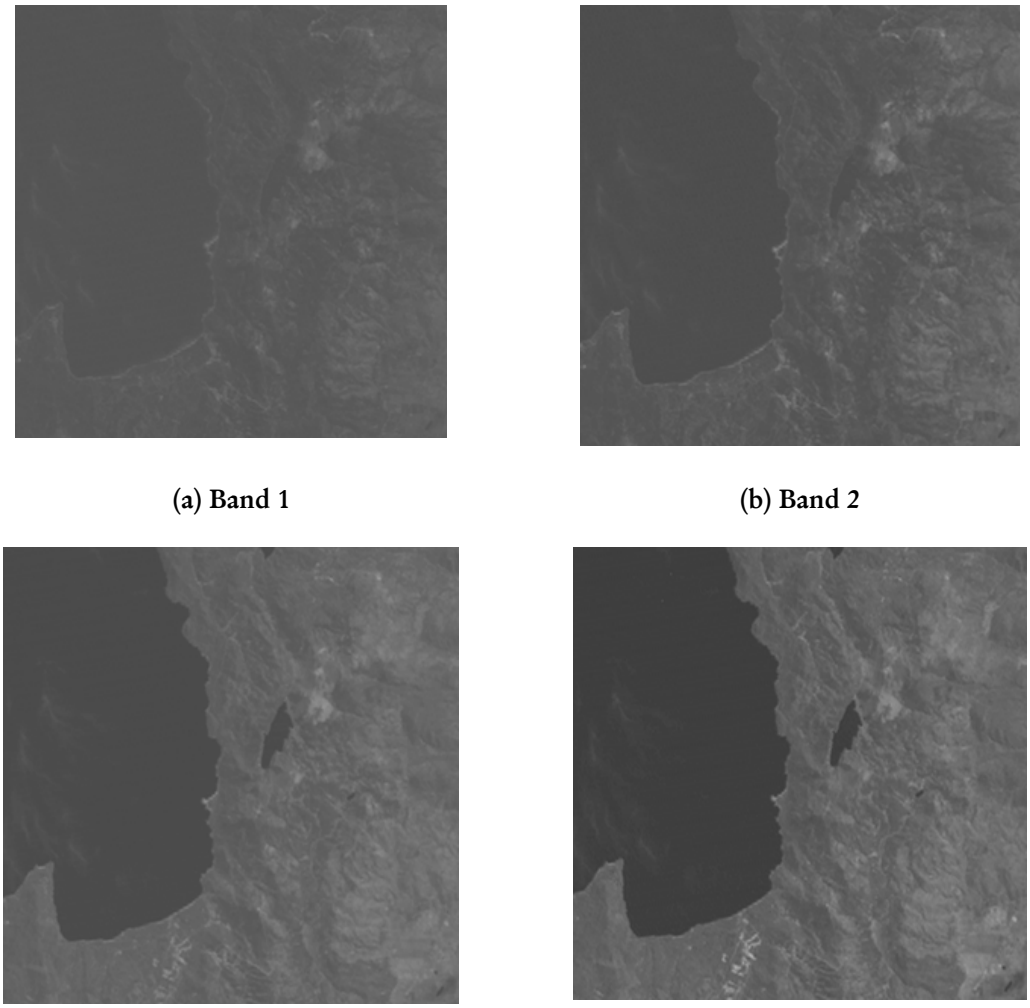
To illustrate the applicability of the proposed approach to multidimensional feature spaces, the PSO-based clustering algorithm was applied to the four-channel multispectral image set of the Lake Tahoe region in the US shown in Figure 4.9. Table 4.14 summarizes the results of applying K-means, *gbest* PSO and *lbest-to-gbest* PSO on the image set. In all the experiments, 50 particles were trained for 100 iterations and V_{max} = 255. All parameters are as for section 4.2.2. The results showed that both PSO approaches performed better than K-means in term of $d_{\min}$. However, *gbest* PSO performed comparably to K-means in terms of $\bar{d}_{\max}$, while *lbest-to-gbest* PSO performed comparably to K-means in terms of J_e. The segmented images (known as thematic maps) for the Tahoe image set are given in Figure 4.10.

Table 4.12: Comparison between K-means, FCM, KHM, H2, GA and PSO for fitness function defined in equation (4.7)

Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	**K-means**	20.21225 ± 0.937836	28.04049 ± 2.7779388	78.4975 ± 7.0628718
	FCM	20.731920 ± 0.650023	28.559214 ± 2.221067	82.434116 ± 4.404686
	KHM(p=2.5)	20.168574 ± 0.0	23.362418 ± 0.0	86.307593 ± 0.000008
	H2 (p=4)	20.136423 ± 0.793973	26.686939 ± 3.011022	81.834143 ± 6.022036
	GA	17.004002 ± 0.035146	24.603018 ± 0.11527	93.492196 ± 0.2567
	PSO	16.988910 ± 0.023937	24.696055 ± 0.130334	93.632200 ± 0.248234
	GCPSO	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954
	PSO noweights	17.284 ± 0.09	22.457 ± 0.414	90.06 ± 0.712
MRI	**K-means**	7.3703 ± 0.042809	13.214369 ± 0.761599	9.93435 ± 7.308529
	FCM	7.205987 ± 0.166418	10.851742 ± 0.960273	19.517755 ± 2.014138
	KHM(p=2.5)	7.53071± 0.129073	10.655988 ± 0.295526	24.270841 ± 2.04944
	H2 (p=4)	7.264114 ± 0.149919	10.926594 ± 0.737545	20.543530 ± 1.871984
	GA	7.038909 ± 0.508953	9.811888 ± 0.419176	25.954191 ± 2.993480
	PSO	7.594520 ± 0.449454	10.186097 ± 1.237529	26.705917 ± 3.008073
	GCPSO	7.555421 ± 0.409742	9.983189 ± 0.915289	27.313118 ± 3.342264
	PSO noweights	7.839 ± 0.238	9.197 ± 0.56	29.45 ± 1.481
Tahoe	**K-means**	3.280730 ± 0.095188	5.234911 ± 0.312988	9.402616 ± 2.823284
	FCM	3.164670 ± 0.000004	4.999294 ± 0.000009	10.970607 ± 0.000015
	KHM(p=2.5)	3.830761 ± 0.000001	6.141770 ± 0.0	13.768387 ± 0.000002
	H2 (p=4)	3.197610 ± 0.000003	5.058015 ± 0.000007	11.052893 ± 0.000012
	GA	3.472897 ± 0.151868	4.645980 ± 0.105467	14.446860 ± 0.857770
	PSO	3.523967 ± 0.172424	4.681492 ± 0.110739	14.664859 ± 1.177861
	GCPSO	3.609807 ± 0.188862	4.757948 ± 0.227090	15.282949 ± 1.018218
	PSO noweights	3.882 ± 0.274	5.036 ± 0.368	16.410 ± 1.231

Table 4.13: Comparison between different non-parametric fitness function				
Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Synthetic	**PSO noweights**	17.284 ± 0.09	22.457 ± 0.414	90.06 ± 0.712
	PSO normalized noweights	17.298567 ± 0.065019	22.387227 ± 0.295405	89.969316 ± 0.482432
MRI	**PSO noweights**	7.839 ± 0.238	9.197 ± 0.56	29.45 ± 1.481
	PSO normalized noweights	7.851594 ± 0.293330	9.182184 ± 0.534796	29.393441 ± 1.240797
Tahoe	**PSO noweights**	3.882 ± 0.274	5.036 ± 0.368	16.410 ± 1.231
	PSO normalized noweights	3.970922 ± 0.218675	5.141907 ± 0.312130	16.746504 ± 1.119426

Table 4.14: Comparison between K-means, *gbest* PSO and *lbest-to-gbest* PSO when applied to multispectral image set				
Image		J_e	$\bar{d}_{\max}$	$d_{\min}$
Four-bands Lake Tahoe	**K-means**	7.281864 ± 0.001512	11.876593 ± 0.001526	17.675578 ± 0.008525
	PSO	8.005989 ± 0.812936	11.935493 ± 0.732004	19.937182 ± 3.468417
	gbest-to-lbest **PSO**	7.639596 ± 0.654930	12.173503 ± 0.740456	18.263982 ± 3.041869



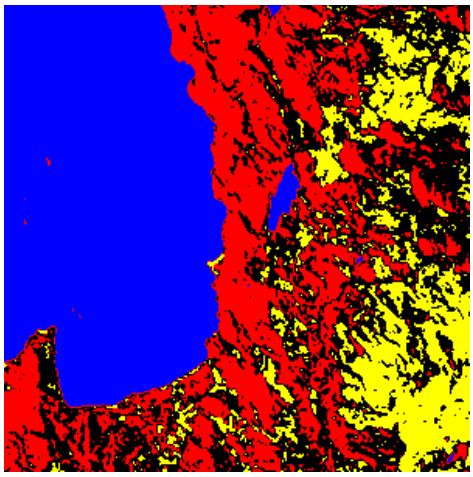
(a) Band 1

(b) Band 2

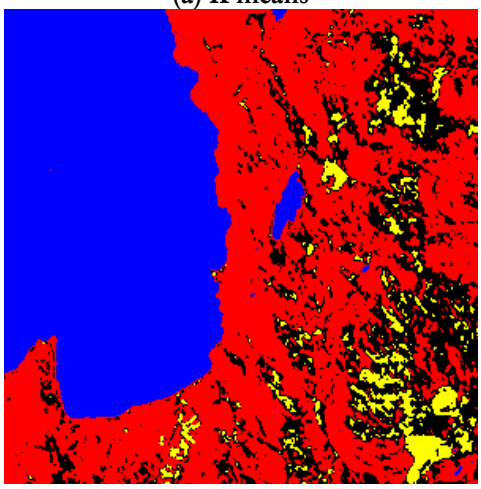
(c) Band 3

(d) Band 4

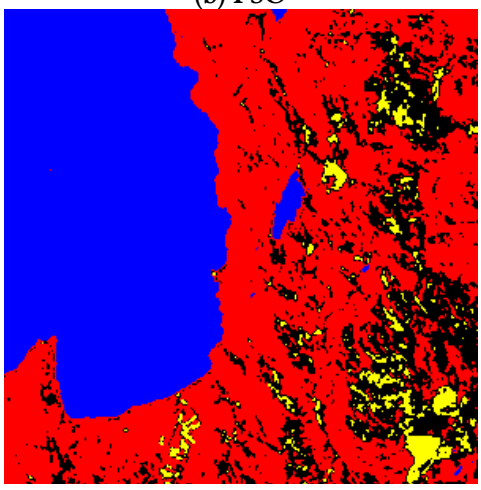
Figure 4.9: The Landsat MSS test images of Lake Tahoe



(a) K-means



(b) PSO



(c) *lbest-to-gbest* PSO

Figure 4.10: The Thematic Maps for Lake Tahoe Image Set

4.2.9 PSO for Data Clustering

The same algorithm presented in section 4.1.1 was used by Van der Merwe and Engelbrecht [2003] to cluster general data sets. It was applied on a set of multi-dimensional data (e.g. the Iris plant data base) using a fitness function consisting of J_e only. In general, the results show that the PSO-based clustering algorithm performs better than the K-means algorithm, which verify the results presented in this chapter. These results are expected since, as previously mentioned, K-means is a greedy algorithm which depends on the initial conditions, which may cause the algorithm to converge to suboptimal solutions. On the other hand, PSO is less sensitive to the effect of the initial conditions due to its population-based nature. Thus, PSO is more likely to find near-optimal solutions.

4.3 Conclusions

This chapter presented a new clustering approach using PSO. The PSO clustering algorithm has as objective to simultaneously minimize the quantization error and intra-cluster distances, and to maximize the inter-cluster distances. Both a *gbest* PSO and GCPSO algorithms have been evaluated. The *gbest* PSO and GCPSO clustering algorithms were further compared against K-means, FCM, KHM, H2 and a GA. In general, the PSO algorithms produced better results with reference to inter- and intra-cluster distances, while having quantization errors comparable to the other algorithms. The performance of different versions of PSO was investigated and the results suggested that algorithms that start with high diversity and then gradually reduces diversity perform better than other algorithms. A non-parametric version of the

proposed fitness function was tested with encouraging results. Finally, the proposed approach was applied to multispectral imagery data.

In the next chapter, a new automatic image generation tool is proposed which is tailored specifically for verification and comparison of different unsupervised image classification algorithms. This tool is used to conduct a more elaborate comparison between the PSO and K-means clustering algorithms.