Chapter 6

Dynamic Clustering using Particle Swarm Optimization with Application to Unsupervised Image Classification

A new dynamic clustering approach (DCPSO), based on PSO, is proposed in this chapter. This approach is applied to unsupervised image classification. The proposed approach automatically determines the "optimum" number of clusters and simultaneously clusters the data set with minimal user interference. The algorithm starts by partitioning the data set into a relatively large number of clusters to reduce the effects of initial conditions. Using binary particle swarm optimization the "best" number of clusters is selected. The centroids of the chosen clusters are then refined via the K-means clustering algorithm. The proposed approach is then applied on synthetic, natural and multispectral images. The experiments conducted show that the proposed approach generally found the "optimum" number of clusters on the tested images. A genetic algorithm and a random search version of dynamic clustering are presented and compared to the particle swarm version.

6.1 The Dynamic Clustering using PSO (DCPSO) Algorithm

This section presents the DCPSO algorithm. For this purpose, define the following symbols:

N_c is the maximum number of clusters.

N_d is the dimension of the data set.

N_p is the number of patterns to be clustered.

$\boldsymbol{Z} = \{z_{j,p} \in \Re \mid j = 1,\ldots, N_d$ and $p = 1,\ldots, N_p\}$ is the set of patterns.

$\boldsymbol{M} = \{m_{j,k} \in \Re \mid j = 1,\ldots, N_d$ and $k = 1,\ldots, N_c\}$ is the set of N_c cluster centroids.

$\boldsymbol{S} = \{\boldsymbol{x}_1,\ldots, \boldsymbol{x}_i,\ldots, \boldsymbol{x}_s\}$ is the swarm of s particles such that $\boldsymbol{x}_i$ indicates particle i, with $x_{i,k} \in \{0,1\}$ for $k = 1,\ldots, N_c$ such that if $x_{i,k} = 1$ then the corresponding centroid $\boldsymbol{m}_k$ in $\boldsymbol{M}$ has been chosen to be part of the solution proposed by particle $\boldsymbol{x}_i$. Otherwise, if $x_{i,k} = 0$ then the corresponding $\boldsymbol{m}_k$ in $\boldsymbol{M}$ is not part of the solution proposed by $\boldsymbol{x}_i$.

n_i is the number of clusters used by the clustering solution represented by particle $\boldsymbol{x}_i$ such that

$$n_i = \sum_{k=1}^{N_c} x_{i,j}\text{ , with } n_i \leq N_c.$$

$\boldsymbol{M}_i$ is the clustering solution represented by particle $\boldsymbol{x}_i$ such that $\boldsymbol{M}_i = (\boldsymbol{m}_k)\ \forall\ k$: $x_{i,k} = 1$ with $\boldsymbol{M}_i \subseteq \boldsymbol{M}$.

n_τ is the number of clusters used by the clustering solution represented by the global best particle $\hat{\boldsymbol{y}}$ (assuming that *gbest* PSO is used) such that

$$n_\tau = \sum_{k=1}^{N_c} \hat{y}_k\text{ , with } n_\tau \leq N_c.$$

$\boldsymbol{M}_\tau$ is the clustering solution represented by $\hat{\boldsymbol{y}}$ such that $\boldsymbol{M}_\tau = (\boldsymbol{m}_k)\ \forall\ k$: $\hat{y}_k = 1$ with $\boldsymbol{M}_\tau \subseteq \boldsymbol{M}$.

$\boldsymbol{M}_r$ is the set of centroids in $\boldsymbol{M}$ which have not been chosen by $\hat{\boldsymbol{y}}$, i.e. $\boldsymbol{M}_r = (\boldsymbol{m}_k)$, $\forall\ k$: $\hat{y}_k = 0$ with $\boldsymbol{M}_r \subseteq \boldsymbol{M}$ (i.e. $\boldsymbol{M}_r \cap \boldsymbol{M}_\tau = \varnothing$ and $\boldsymbol{M}_r \cup \boldsymbol{M}_\tau = \boldsymbol{M}$).

p_{ini} is a user-specified probability defined in Kuncheva and Bezdek [1998], which is used to initialize a particle position, $\boldsymbol{x}_i$, as follows:

$$x_{i,k}(t) = \begin{cases} 0 & \text{if } r_k(t) \geq p_{\text{ini}} \\ 1 & \text{if } r_k(t) < p_{\text{ini}} \end{cases} \tag{6.1}$$

where $r_k(t) \sim U(0,1)$. Obviously a large value for p_{ini} results in selecting most of the centroids in $\boldsymbol{M}$.

The algorithm which uses some of the ideas presented by Kuncheva and Bezdek [1998]: A pool of cluster centroids, $\boldsymbol{M}$, is randomly chosen from $\boldsymbol{Z}$. The swarm of particles, $\boldsymbol{S}$, is then randomly initialized. Binary PSO is then applied to find the "best" set of cluster centroids, $\boldsymbol{M}_\tau$, from $\boldsymbol{M}$. K-means is applied to $\boldsymbol{M}_\tau$ in order to refine the chosen centroids. $\boldsymbol{M}$ is then set to $\boldsymbol{M}_\tau$ plus $\boldsymbol{M}_r$, which is a randomly chosen set of centroids from $\boldsymbol{Z}$ (this is done to add diversity to $\boldsymbol{M}$ and to reduce the effect of the initial conditions). The algorithm is then repeated using the new $\boldsymbol{M}$. When the termination criteria are met, $\boldsymbol{M}_\tau$ will be the resulting "optimum" set of cluster centroids and n_τ will be the "optimum" number of clusters in $\boldsymbol{Z}$. The DCPSO algorithm is summarized in Figure 6.1.

The termination criterion can be a user-defined maximum number of iterations or a lack of progress in improving the best solution found so far for a user-specified consecutive number of iterations, TC. In this chapter, the latter approach is used with $TC_1 = 50$ for Step 6 and $TC_2 = 2$ for step 10. These values for TC were set empirically. N_c and s are user defined parameters.

```
1)  Select $\boldsymbol{m}_k \in \boldsymbol{M}$, $\forall\, k = 1,\ldots, N_c$ where $1 < N_c < N_p$, randomly from $\boldsymbol{Z}$
2)  Initialize the swarm $\boldsymbol{S}$, with $x_{i,k} \sim U\{0,1\}$, $\forall\, i = 1,\ldots, s$ and $k = 1,\ldots, N_c$ using equation (6.1)
3)  Randomly initialize the velocity, $\boldsymbol{v}_i$, of each particle $i$ in $\boldsymbol{S}$ such that $v_{i,k} \in [-5,5]$, $\forall\, i = 1,\ldots, s$ and $k = 1,\ldots, N_c$. The range of [-5,5] was set empirically
4)  For each particle $\boldsymbol{x}_i$ in $\boldsymbol{S}$
        a. Partition $\boldsymbol{Z}$ according to the centroids in $\boldsymbol{M}_i$ by assigning each data point $\boldsymbol{z}_p$ to the closest (in terms of the Euclidean distance) cluster in $\boldsymbol{M}_i$
        b. Calculate the clustering validity index, $VI$, using one of the clustering validity indices as defined in section 3.1.4 to measure the quality of the resulting partitioning of $\boldsymbol{Z}$ (i.e. $VI = V$, $VI = S\_Dbw$ or $VI=1/D$ since $D$ should be maximized)
        c. $f(\boldsymbol{x}_i) = VI$
5)  Apply the binary PSO velocity and position update equations (2.8) and (2.15) on all particles in $\boldsymbol{S}$
6)  Repeat steps 4) and 5) until a termination criterion is met
7)  Adjust $\boldsymbol{M}_\tau$ by applying the K-means clustering algorithm
8)  Randomly re-initialize $\boldsymbol{M}_r$ from $\boldsymbol{Z}$
9)  Set $\boldsymbol{M} = \boldsymbol{M}_r \cup \boldsymbol{M}_\tau$
10) Repeat steps 2) through 9) until a termination criterion is met
```

Figure 6.1: The DCPSO algorithm

A GA version of DCPSO can easily be implemented by replacing step 5 in the above algorithm with GA evolutionary operators for selection, crossover and mutation. On the other hand, a random search (RS) version of DCPSO, as described by Kunchevea and Bezdek [1998], can be implemented by removing step 5 and keeping only the best solution encountered so far.

As an illustration of the DCPSO algorithm, consider the following example.

Example 6.1

Let $N_p = 100$, $N_d = 1$, and $N_c = 5$.

Let $\boldsymbol{M}$ be

3	29	78	150	200

An example of a particle position, $\boldsymbol{x}_i$, is

0	1	1	0	1

which means that cluster centers 29, 78 and 200 are chosen for this particle such that $\boldsymbol{M}_i$ is

29	78	200

In other words, all data in $\boldsymbol{Z}$ are grouped in only these three clusters.

After step 6, assume the global best particle, $\hat{\boldsymbol{y}}$, is

0	1	0	1	1

Then, $\boldsymbol{M}_\tau$ is

29	150	200

Assume that after K-means is applied on $\boldsymbol{Z}$ using the centroids given by $\boldsymbol{M}_{\tau}$, the new $\boldsymbol{M}_{\tau}$ is given by

30.5	129.9	201

Then, randomly initialize the remaining N_c - n_{τ} (i.e. 5 - 3 = 2) clusters, representing $\boldsymbol{M}_{r}$, from $\boldsymbol{Z}$ (shown below in bold). The resultant $\boldsymbol{M}$ may look as follows:

110	30.5	**8**	129.9	201

The DCPSO algorithm is then repeated using the new $\boldsymbol{M}$.

6.1.1 Validity Index

One of the advantages of DCPSO is that it can use any validity index. Therefore, the user can choose the validity index suitable for his/her data set. In addition, any new index can easily be integrated with DCPSO. The validity indices used in this chapter are *D*, *V* and *S_Dbw* (as defined in section 3.1.4).

6.1.2 Time Complexity

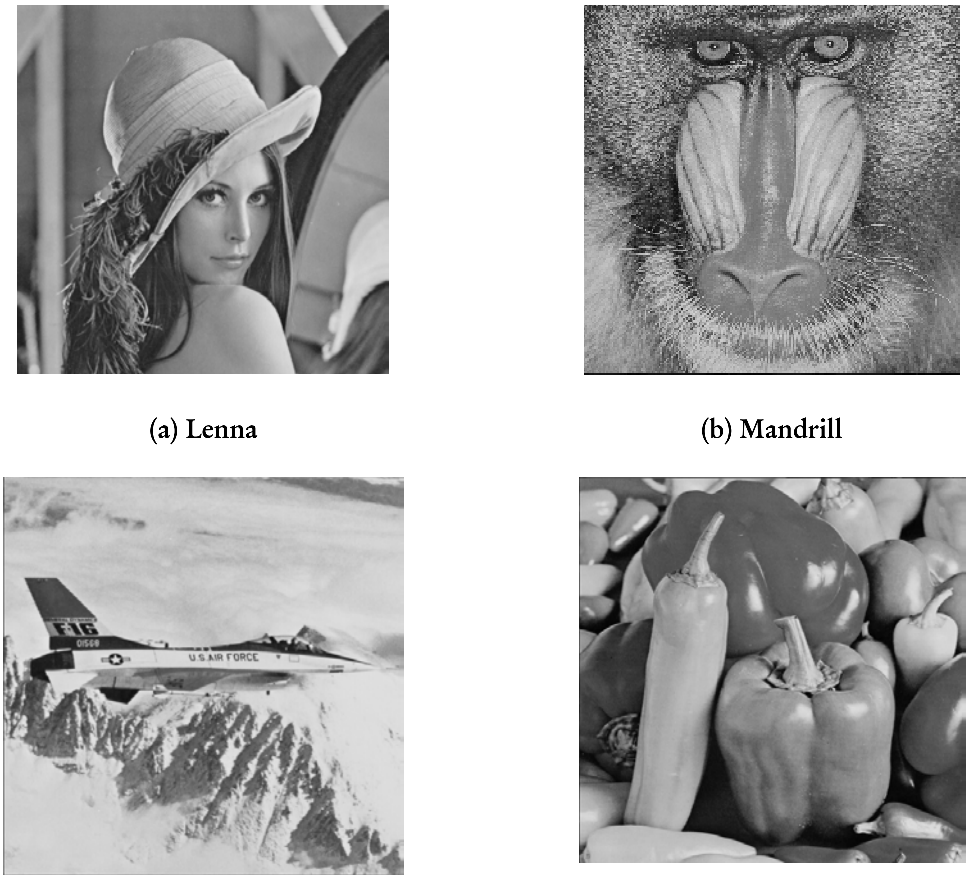
The time complexity of DCPSO is based on the complexity of four processes, namely, the partitioning of $\boldsymbol{Z}$, calculating the quality of the partition, applying binary PSO and applying K-means. Assume that T_1 is the number of iterations taken by the PSO to converge (step 6 of the algorithm), and that T_2 is the number of iterations taken by DCPSO to converge (step 10 of the algorithm). Then the complexity of partitioning $\boldsymbol{Z}$ is O($sT_1T_2N_cN_pN_d$), while the complexity of calculating the quality of a partition will depend on the time complexity of the validity index which is, in general, some constant, ξ, multiplied by N_p for the indices used in this chapter. The complexity of this step is therefore O($\xi T_1T_2N_p$). Finally, the complexity of K-means is $O(N_p)$. The parameters T_1, T_2, N_c, s and ξ can be fixed in advance. Typically, T_1, T_2, N_c, s, ξ, N_d << N_p. Let ς be the multiplication of s, T_1, T_2, N_c and N_d (i.e. $\varsigma = sT_1T_2N_cN_d$). If $\varsigma << N_p$ then the time complexity of DCPSO will be $O(N_p)$. However, if $\varsigma \approx N_p$ then the time complexity of DCPSO will be O(N_p^2).

6.2 Experimental results

Experiments were conducted using both synthetic images and natural images. The synthetic images were generated by SIGT as given in Table 5.1. Furthermore, SIGT was used to generate another five different synthetic images for which the actual number of clusters was known in advance. These images have different numbers of clusters with varying complexities; they consist of well separated clusters, overlapping clusters or a combination of both. The new five synthetic images are given in Table 6.1 along with their histograms.

The following well known natural images were used: *Lenna*, *mandrill*, *jet* and *peppers*. These images are shown in Figure 6.2. Furthermore, one MRI and one satellite image of Lake Tahoe (as given in Figure 4.2) have been used to show the wide applicability of the proposed approach.

The remainder of this section is organized as follows: Section 6.2.1 applies DCPSO to the synthetic images using the three validity indices described in section 6.1.1. These results are compared with the unsupervised fuzzy approach (UFA) proposed by Lorette *et al.* [2000] (discussed in section 3.1.5) and the SOM approach (refer to section 3.1.6). In section 6.2.2, the same experiments are conducted on the natural images. Section 6.2.3 compares DCPSO with GA and RS versions on the natural images. Sections 6.2.4, 6.2.5 and 6.2.6 investigate the influence of the different DCPSO control parameters. Different PSO models (namely, *lbest*, *gbest* and *lbest-to-gbest*) are investigated in section 6.2.7. Finally, section 6.2.8 applies DCPSO to multispectral imagery data.

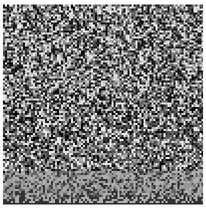
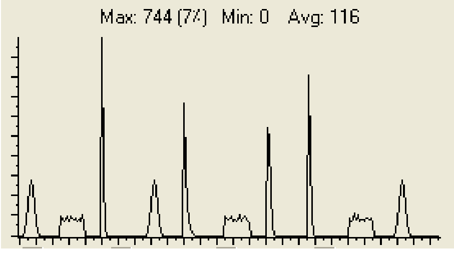
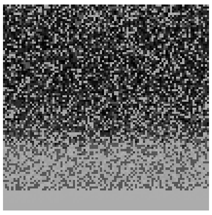
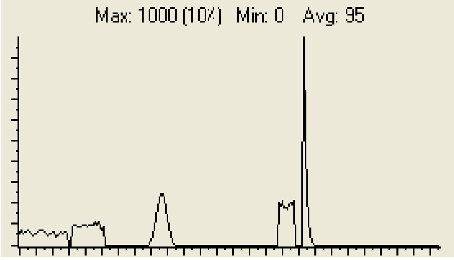
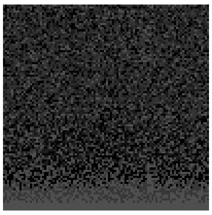
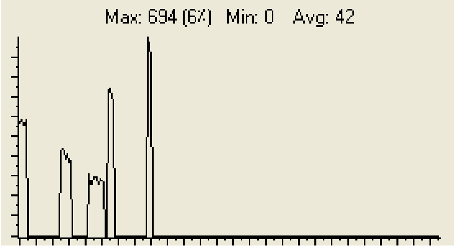
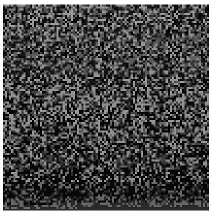
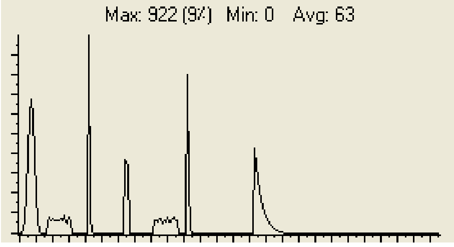
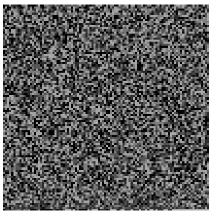
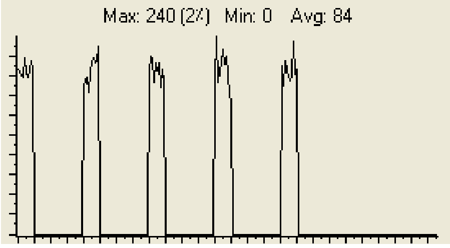


(a) Lenna **(b) Mandrill**

(c) Jet **(d) Peppers**

Figure 6.2: Natural Images

Table 6.1: Additional synthetic images used along with the corresponding histograms

Synthetic image no.	Image	Histogram
11		
12		
13		
14		
15		

The results reported in this section are averages and standard deviations over 20 simulations. Since *lbest-to-gbest* PSO was generally the best performer in chapter 4, *lbest-to-gbest* PSO is used in this section unless otherwise specified. Furthermore, if the best solution has not been improved after a user-specified number of iterations (50 iterations was used for all the experiments conducted) then the algorithm was terminated (Step 6 of the algorithm, Section 6.1). For the index proposed by Turi [2001], parameter c was set to 25 in all experiments as recommended by by Turi [2001]. The DCPSO parameters were empirically set as follows: N_c = 20, p_{ini} = 0.75 and s = 100 for all experiments conducted unless otherwise specified. In addition, the PSO parameters were set as follows: w =0.72, $c_1 = c_2$ = 1.49 and $V_{\max}$= 255. For UFA, the user-defined parameter, ε, was set equal to $1/N_p$ as suggested by Lorette *et al.* [2000]. For the SOM, a Kohonen network of 5×4 nodes was used (to give a minimum of 20 codebook vectors). All implementation issues were set as in Pandya and Macy [1996]: the learning rate $\eta(t)$ was initially set to 0.9 then decreased by 0.005 until it reached 0.005; the neighborhood function $\Delta_w(t)$ was initially set to (5+4)/4 then decreased by 1 until it reached zero.

6.2.1 Synthetic images

Table 6.2 summarizes the results of DCPSO using the three validity indices described in section 6.1.1, along with the UFA and SOM results. It appears that UFA tends to overfit the data since it selected the maximum number of clusters as the correct one for all experiments. The rationale behind this failure is the choice of ε which has a significant effect on the resulting number of clusters. DCPSO using *S_Dbw* also generally overfits the data. On the other hand, DCPSO using D, DCPSO using V and

SOM have generally performed very well (especially DCPSO using *V*). Hence, it can be concluded that DCPSO using *V* is efficient with respect to the synthetic images.

Table 6.2: Experiments on synthetic images

Image	Actual no. of clusters	DCPSO using *D*	DCPSO using *V*	DCPSO using *S_Dbw*	SOM	UFA
1	2	2 ± 0	2 ± 0	5.55 ± 5.22	2	20
2	3	3 ± 0	3 ± 0	3 ± 0	3	20
3	3	2 ± 0	2 ± 0	4.4 ± 4.852	6	20
4	3	2.7 ± 1.345	5.15 ± 0.357	10.9 ± 5.458	10	20
5	4	10.85 ± 1.878	5 ± 0	15.5 ± 1.323	7	20
6	10	9.55 ± 2.246	7.2 ± 0.872	9.3 ± 0.458	9	20
7	6	3.35 ± 1.526	7.9 ± 0.995	10.8 ± 2.925	9	20
8	4	8.8 ± 2.379075	5 ± 0	4 ± 0	4	20
9	7	4.25 ± 0.433	5 ± 0	14.1 ± 3.52	13	20
10	4	7.9 ± 1.729	7 ± 0	13.95 ± 1.77	9	20
11	10	10.0 ± 0.950	10 ± 0	10 ± 0	10	20
12	5	9.0 ± 2.168	7.2 ± 0.4	11.65 ± 1.06	6	20
13	5	12.1 ± 2.119	5 ± 0	9.6 ± 2.107	5	20
14	7	7.5 ± 1.204	5 ± 0	7.8 ± 2.088	7	20
15	5	5 ± 0	5 ± 0	5 ± 0	5	20
Avg.	**5.2**	**6.53**	**5.43**	**9.04**	**7**	**20**

6.2.2 Natural images

Table 6.3 shows the results of DCPSO using the three validity indices described in section 6.1.1. These results are compared with the results of UFA and SOM. In addition, the results of snob for the Lenna, mandrill, jet and peppers images are copied from Turi [2001]. The optimal range for the number of clusters for the images of Lenna, mandrill, jet and peppers are also taken from Turi [2001] which was based on a visual analysis survey conducted by a group of ten people. Similarly, the optimal range for the MRI and Lake Tahoe images were estimated using a group of three people. It appears from the table that results of DCPSO using *S_Dbw*, UFA, SOM and snob were poor. DCPSO using *V* always found a solution within the optimal range. Therefore, the remaining experiments will use *V* as the validity index. These results clearly show the efficiency of DCPSO. Table 6.4 provides samples of the resultant segmented images generated by DCPSO using *V*.

Table 6.3: Experiments on natural images

Image	Optimal range	DCPSO using *D*	DCPSO using *V*	DCPSO using *S_Dbw*	SOM	UFA	Snob
Lenna	5 to 10	10.35 ± 1.652	6.85 ± 0.477	19.3 ± 0.843	20	20	31
Mandrill	5 to 10	6.05 ± 1.658	6.25 ± 0.433	19.25 ± 0.766	20	20	42
Jet	5 to 7	3.35 ± 2.151	5.3 ± 0.459	18.05 ± 1.465	14	20	22
peppers	6 to 10	10.55 ± 1.465	6 ± 0	18.8 ± 0.872	20	20	39
MRI	4 to 8	3 ± 0	5 ± 0	17.2 ± 1.4	19	20	-
Tahoe	3 to 7	3 ± 0	6.1 ± 0.539	14.3 ± 3.018	4	20	-
Avg.		**6.05**	**5.92**	**17.82**	**16.17**	**20**	**-**

Table 6.4: Samples of segmented images resulting from DCPSO using *V*

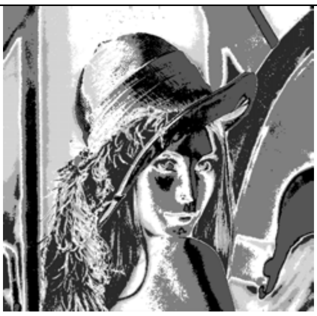
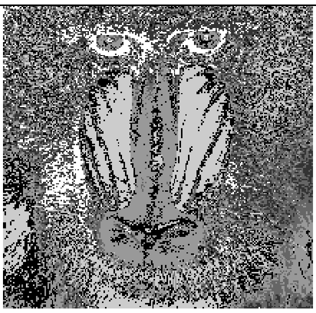
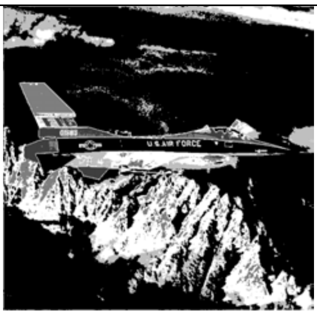
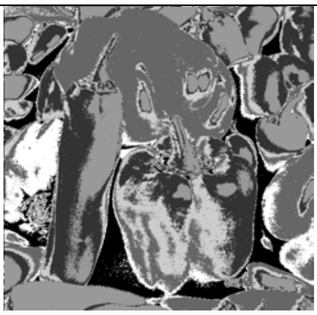
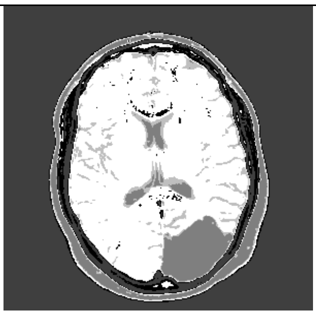
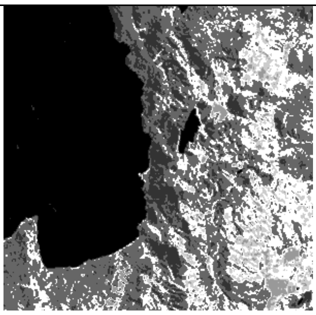
Image	Segmented image	No. of clusters
Lenna		7
Mandrill		6
Jet		5

Table 6.4: Samples of segmented images resulting from DCPSO using *V* (*continued*)

Image	Image	Image
peppers		6
MRI		5
Tahoe		6

6.2.3 Comparison with GA and RS

The previous experiments were conducted using the dynamic cluster PSO. In this section, a GA and RS version of the algorithm in Figure 6.1 (called DCGA and DCRS, respectively) are examined and compared with DCPSO. Both DCGA and DCRS used 100 individuals. For DCGA, elitism was used, keeping the fittest chromosome for the next generation. In addition, random selection has been used along with uniform crossover. The crossover probability was set to 0.8 with a mixing ratio of 0.5; a mutation probability of $1/N_c$ was used. Table 6.5 presents the results of applying DCPSO, DCGA and DCRS on the natural images. As expected, DCRS performed poorly due to its pure random search. DCGA performed comparably to DCPSO.

Table 6.5: Comparison of PSO-, GA- and RS- versions of the proposed approach

Image	Optimal range	DCPSO using V	DCGA using V	DCRS using V
Lenna	5 to 10	6.85 ± 0.477	6.45 ± 0.74	9.8 ± 1.661
Mandrill	5 to 10	6.25 ± 0.433	6.05 ± 0.589	8.75 ± 2.095
Jet	5 to 7	5.3 ± 0.459	5.3 ± 0.557	11.05 ± 1.627
peppers	6 to 10	6 ± 0	6.05 ± 0.218	10.55 ± 1.532
MRI	4 to 8	5 ± 0	5.5 ± 0.742	8.1 ± 1.179
Tahoe	3 to 7	6.1 ± 0.539	6.1 ± 0.831	9.25 ± 1.479
Avg.		**5.92**	**5.91**	**9.58**

6.2.4 Swarm Size

Reducing the swarm size (or population size in case of GA) from 100 to 20 particles (or GA chromosomes) did not generally affect the performance of either DCPSO or DCGA as illustrated in Table 6.6. However, comparing Table 6.5 and Table 6.6 it seems that on average less clusters are formed with less particles/chromosomes. In general, the computational requirements of DCPSO and DCGA can be reduced significantly without affecting the performance of DCPSO and DCGA.

Table 6.6: Comparison of PSO- and GA-versions of the proposed approach using a swarm size $s = 20$

Image	Optimal range	DCPSO using V	DCGA using V
Lenna	5 to 10	6.5 ± 0.806	6.4 ± 0.8
Mandrill	5 to 10	6.15 ± 0.357	5.85 ± 0.476
Jet	5 to 7	5.3 ± 0.458	5.35 ± 0.477
peppers	6 to 10	6.05 ± 0.218	6 ± 0
MRI	4 to 8	5.2 ± 0.4	5.15 ± 0.357
Tahoe	3 to 7	6.05 ± 0.384	6.2 ± 0.4
Avg.		**5.875**	**5.825**

6.2.5 The Termination Criteria

Given that all parameters are fixed at the values given in section 6.2.4, the influence of the termination criteria were evaluated for the natural images. The termination criterion for step 6 in the algorithm (section 6.1) is called TC_1 and for step 10 is called TC_2.

Table 6.7 and 6.8 summarize the effect of TC_1 and TC_2, respectively. In Table 6.7, TC_2 was fixed at 2. Table 6.7 shows that for the Lenna, Mandrill and MRI images all the tested values for TC_1 produced comparable results within the optimal range. For the Jet image, all the tested values for TC_1 performed comparably with TC_1=5 and TC_1=25 slightly worse than the other values. For the Peppers image, TC_1=75 and TC_1=100 performed better than other values and the results suggest that the "optimal" number of clusters in the Pepper image is 6 which seems to be a valid number. For the Tahoe image, all the test values for TC_1 (except TC_1=75) produced comparable results within the optimal range. From the results shown in Table 6.7, it can be concluded that the performance of DCPSO is generally insensitive to TC_1's values.

In Table 6.8, TC_1 was fixed at 50. Table 6.8 shows that for the Lenna and Peppers images all the values of TC_2 (except for TC_2=2) produced the "optimal" number of clusters. For the Mandrill image, all the tested values for TC_2 produced comparable results within the optimal range. For the Jet image, all the values of TC_2 (expect TC_2=2) produced results within the optimal range. TC_2=25 and TC_2=50 suggest that the "optimal" number of clusters in the MRI image is 5. This seems to be a valid number since the Brain MRI images consist mainly of three major tissue classes: gray matter, white matter and cerebrospinal fluid [Zhang *et al*. 2001]. Furthermore, the images contain the skull and the background. For the Tahoe image, TC_2=25 and TC_2=50 produced results outside the optimal range. From the results shown in Table 6.8, it can be concluded that the performance of DCPSO is relatively insensitive to TC_2's values.

Table 6.7: Effect of termination criterion TC_1 on the DCPSO using a swarm size $s = 20$ and $TC_2 = 2$

Image	TC_1	Optimal range	DCPSO using V
Lenna	5	5 to 10	6.05 ± 0.921
	25	5 to 10	6.55 ± 0.740
	50	5 to 10	6.5 ± 0.806
	75	5 to 10	6.55 ± 0.740
	100	5 to 10	6.55 ± 0.805
Mandrill	5	5 to 10	6.05 ± 1.023
	25	5 to 10	6.1 ± 0.539
	50	5 to 10	6.15 ± 0.357
	75	5 to 10	5.95 ± 0.384
	100	5 to 10	6.05 ± 0.218
Jet	5	5 to 7	5.35 ± 0.726
	25	5 to 7	5.2 ± 0.4
	50	5 to 7	5.3 ± 0.458
	75	5 to 7	5.35 ± 0.477
	100	5 to 7	5.35 ± 0.477
Peppers	5	6 to 10	6.45 ± 1.023
	25	6 to 10	6.2 ± 0.678
	50	6 to 10	6.05 ± 0.218
	75	6 to 10	6.0 ± 0.0
	100	6 to 10	6.0 ± 0.0
MRI	5	4 to 8	5.7 ± 0.843
	25	4 to 8	5.25 ± 0.698
	50	4 to 8	5.2 ± 0.4
	75	4 to 8	5.1 ± 0.3
	100	4 to 8	5.15 ± 0.477
Tahoe	5	3 to 7	5.85 ± 0.477
	25	3 to 7	6.15 ± 0.572
	50	3 to 7	6.05 ± 0.384
	75	3 to 7	6.45 ± 0.669
	100	3 to 7	6.2 ± 0.4

Table 6.8: Effect of termination criterion TC_2 on the DCPSO using a swarm size s = 20 and TC_1= 50

Image	TC_2	Optimal range	DCPSO using V
Lenna	2	5 to 10	6.55 ± 0.740
	10	5 to 10	7.0 ± 0.0
	25	5 to 10	7.0 ± 0.0
	50	5 to 10	7.0 ± 0.0
Mandrill	2	5 to 10	6.1 ± 0.539
	10	5 to 10	6.25 ± 0.433
	25	5 to 10	6.15 ± 0.357
	50	5 to 10	6.15 ± 0.357
Jet	2	5 to 7	5.2 ± 0.4
	10	5 to 7	5.6 ± 0.49
	25	5 to 7	5.6 ± 0.49
	50	5 to 7	5.8 ± 0.4
Peppers	2	6 to 10	6.2 ± 0.678
	10	6 to 10	6.0 ± 0.0
	25	6 to 10	6.0 ± 0.0
	50	6 to 10	6.0 ± 0.0
MRI	2	4 to 8	5.25 ± 0.698
	10	4 to 8	5.05 ± 0.218
	25	4 to 8	5.0 ± 0.0
	50	4 to 8	5.0 ± 0.0
Tahoe	2	3 to 7	6.15 ± 0.572
	10	3 to 7	6.4 ± 0.49
	25	3 to 7	6.75 ± 0.829
	50	3 to 7	6.85 ± 0.792

6.2.6 p_{ini} and N_c

Given that all parameters are fixed at the values given in section 6.2.4, the influence of p_{ini} was evaluated for the natural images. The results are summarized in Table 6.9. Studying the results, it can be concluded that the performance of DCPSO is generally insensitive to the value of p_{ini}.

Table 6.9: Effect of p_{ini} on the DCPSO using a swarm size s = 20

Image	p_{ini}	Optimal range	DCPSO using V
Lenna	0.25	5 to 10	6.7 ± 0.64
	0.5	5 to 10	6.5 ± 0.742
	0.75	5 to 10	6.5 ± 0.806
	0.9	5 to 10	6.65 ± 0.726
Mandrill	0.25	5 to 10	6.05 ± 0.218
	0.5	5 to 10	6.05 ± 0.21
	0.75	5 to 10	6.15 ± 0.357
	0.9	5 to 10	6.1 ± 0.539
Jet	0.25	5 to 7	5.5 ± 0.592
	0.5	5 to 7	5.3 ± 0.458
	0.75	5 to 7	5.3 ± 0.458
	0.9	5 to 7	5.3 ± 0.458
Peppers	0.25	6 to 10	6.0 ± 0.0
	0.5	6 to 10	6.0 ± 0.0
	0.75	6 to 10	6.05 ± 0.218
	0.9	6 to 10	6.0 ± 0.0
MRI	0.25	4 to 8	5.3 ± 0.781
	0.5	4 to 8	5.35 ± 0.726
	0.75	4 to 8	5.2 ± 0.4
	0.9	4 to 8	5.3 ± 0.9
Tahoe	0.25	3 to 7	6.2 ± 0.51
	0.5	3 to 7	6.35 ± 0.477
	0.75	3 to 7	6.05 ± 0.384
	0.9	3 to 7	6.05 ± 0.497

Given that all parameters are fixed at the values given in section 6.2.4, the influence of N_c was evaluated for the natural images. The results are summarized in Table 6.10. Studying the results, it appears that using N_c = 10 generally results in choosing the lower bound of the optimal range. However, using N_c = 50 tends to overfit the data by producing results outside the optimal range. Table 6.10 shows that, N_c = 20 generates the best results for the natural images. Hence, it can be concluded that the performance of DCPSO is sensitive to the value of N_c.

Table 6.10: Effect of N_c on the DCPSO using a swarm size $s = 20$

Image	N_c	Optimal range	DCPSO using V
Lenna	10	5 to 10	5.4 ± 0.583
	20	5 to 10	6.5 ± 0.806
	50	5 to 10	16.8 ± 3.516
Mandrill	10	5 to 10	5.55 ± 0.497
	20	5 to 10	6.15 ± 0.357
	50	5 to 10	15.95 ± 3.57
Jet	10	5 to 7	5.05 ± 0.218
	20	5 to 7	5.3 ± 0.458
	50	5 to 7	15.35 ± 2.495
Peppers	10	6 to 10	5.9 ± 0.3
	20	6 to 10	6.05 ± 0.218
	50	6 to 10	16.7 ± 2.722
MRI	10	4 to 8	5.25 ± 0.433
	20	4 to 8	5.2 ± 0.4
	50	4 to 8	11.45 ± 3.892
Tahoe	10	3 to 7	5.35 ± 0.572
	20	3 to 7	6.05 ± 0.384
	50	3 to 7	12.9 ± 4.265

6.2.7 Comparison of *gbest*-, *lbest*- and *lbest-to-gbest*-PSO

In this section, the effect of different models of PSO is investigated. A comparison is made between *gbest*-, *lbest*- and *lbest-to-gbest*-PSO (which has been used in the above experiments) using a swarm size of 20 particles. For *lbest*-PSO, a neighborhood size of l = 2 was used. Table 6.11 summarizes the result of the comparison. The results show no significant difference in performance.

Table 6.11: Comparison of *gbest-*, *lbest-* and *lbest-to-gbest-* PSO versions of DCPSO using V (s = 20)

Image	Optimal range	*gbest-* PSO	*lbest* PSO (l=2)	*lbest-to-gbest-* PSO
Lenna	5 to 10	6.6 ± 0.735	6.55 ± 0.669	6.5 ± 0.806
Mandrill	5 to 10	6.1 ± 0.3	6.1 ± 0.539	6.15 ± 0.357
Jet	5 to 7	5.5 ± 0.5	5.25 ± 0.433	5.3 ± 0.458
peppers	6 to 10	6.15 ± 0.726	6.15 ± 0.654	6.05 ± 0.218
MRI	4 to 8	5.0 ± 0.0	5.3 ± 0.458	5.2 ± 0.4
Tahoe	3 to 7	6.25 ± 0.829	6.1 ± 0.3	6.05 ± 0.384
Avg.		**5.933**	**5.908**	**5.875**

6.2.8 Multispectral Imagery Data

To show the applicability of DCPSO to multidimensional feature spaces, DCPSO was applied to the four-channel multispectral image set of the Lake Tahoe region in the US. The four bands of the image set was already shown in Figure 4.9. Table 6.12 gives the results of applying *lbest-to-gbest* DCPSO using V on the image set. The results reported in Table 6.12 are averages and standard deviations over 10 simulations. All parameters are fixed at the values given in section 6.2.4. It appears from the table that DCPSO using V found a solution within the optimal range. The results show the efficiency of DCPSO when applied to multispectral imagery data. Figure 6.3 shows a sample of the resultant segmented image (or thematic map) generated by DCPSO using V.

Table 6.12: Applying *lbest-to-gbest* DCPSO using *V* (*s* = 20) on multispectral image set		
Image	Optimal range	DCPSO using *V*
Four-bands Lake Tahoe	3 to 7	5.8 ± 0.6

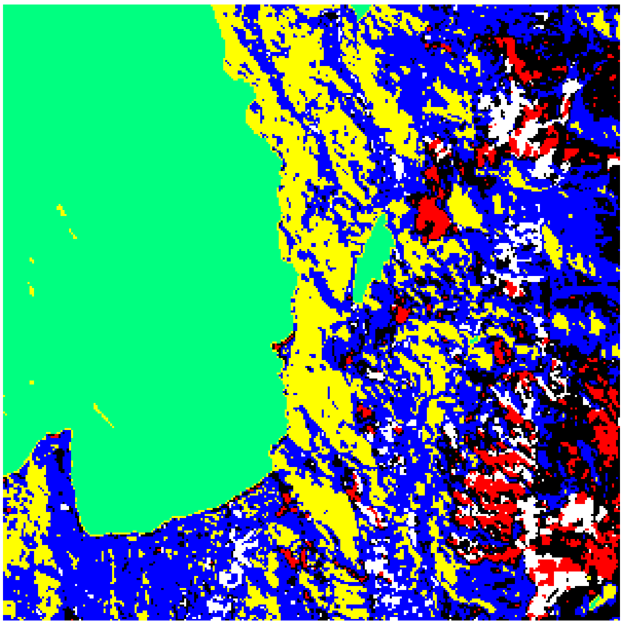


Figure 6.3: 6-Clusters thematic map obtained using DCPSO

6.3 Conclusions

This chapter presented DCPSO, a new dynamic clustering algorithm based on PSO with application to unsupervised image classification. DCPSO clusters a data set without requiring the user to specify the number of clusters in advance. This is an important feature since knowing the number of clusters in advance is often not easy. DCPSO uses a validity index to measure the quality of the resultant clustering. One of the advantages of this approach is that DCPSO can work with any validity index. In addition, the proposed approach can be used with a GA or RS. DCPSO was applied on synthetic (where the number of clusters was known in advance) as well as natural images (including MRI and satellite images), and was compared with other unsupervised clustering techniques. From these experiments it can be concluded that DCPSO, using the validity index proposed by Turi [2001], has outperformed other approaches. In general, DCPSO successfully found the optimum number of clusters on the tested images. DCPSO was then compared to both DCGA and DCRS, with DCPSO and DCGA outperforming DCRS. The influence of the different DCPSO control parameters was then investigated. The use of different PSO models (namely, *lbest*, *gbest* and *lbest-to-gbest*) was also studied. Finally, DCPSO was successfully applied to multispectral imagery data.

The next chapter applies the PSO clustering approach to two difficult problems in the fields of pattern recognition and image processing, namely, color image quantization and spectral unmixing.