10

Swarm Intelligence and Image Segmentation

Sara Saatchi and Chih-Cheng Hung
Southern Polytechnic State University
USA

1. Introduction

Image segmentation plays an essential role in the interpretation of various kinds of images. Image segmentation techniques can be grouped into several categories such as edge-based segmentation, region-oriented segmentation, histogram thresholding, and clustering algorithms (Gonzalez & Woods, 1992). The aim of a clustering algorithm is to aggregate data into groups such that the data in each group share similar features while the data clusters are being distinct from each other.

The K-means algorithm is a widely used method used for finding the structure of data (Tou & Gonzalez 1974). This unsupervised clustering technique has a strong tendency to get stuck into local minima when finding an optimal solution. Therefore, clustering results are heavily dependent on the initial cluster centers distribution. Hence, the search for good initial parameters is a challenging issue and the clustering algorithms require a great deal of experimentation to determine the input parameters for the optimal or suboptimal clustering results.

Competitive learning model introduced in (Rumelhart & Zipser, 1986) is an interesting and powerful learning algorithm which can be used in unsupervised training for image classification (Hung, 1993). Simple Competitive Learning (SCL), shows stability over different run trials but this stable result is not always the global optima. In fact, in some cases the SCL converges to local optima over all run trials and the learning rate needs to be adjusted in the course of experimentation so that the global optimization can be achieved.

There are a number of techniques, developed for optimization, inspired by the behaviours of natural systems (Pham & Karaboga, 2000). Swarm intelligence (SI) including Ant Colony Optimization (ACO) introduced in (Dorigo et al., 1996) and Particle Swarm Optimization (PSO) introduced in (Kennedy & Eberhart, 1995) has been introduced in the literature as an optimization technique. There are several SI approaches for data clustering in the literature which use clustering techniques such as K-means algorithm. In most of these approaches ACO or PSO are used to obtain the initial cluster centers for the K-means algorithm. We propose a hybrid algorithm which combines SI with K-means. We also use the same method to combine SI with SCL.

Our aim is to make segmentation results of both K-means and SCL less dependent on the initial cluster centers and learning rate respectively. Hence, their results are more accurate and stabilized by employing the ACO and PSO optimization techniques. This improvement is due to the larger search space provided by these techniques. In addition, our

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

methodology of considering both spatial and spectral features of the image helps to produce results with improved accuracy.

We have integrated the K-means and simple competitive learning algorithms with ACO in (Saatchi & Hung, 2005) and (Saatchi & Hung, 2006) respectively. In this paper we will study the hybridization of PSO with each of the K-means and the SCL algorithms. A thorough comparison study on ACO-K-means, PSO-K-means, ACO-SCL, PSO-SCL, K-means and SCL algorithms will also be provided.

2. Clustering Algorithms

2.1 K-means

The K-means algorithm, first introduced in (McQueen, 1967), is an is an unsupervised clustering algorithm which partitions a data set into a certain number of clusters. The K-means algorithm is based on the minimization of a performance index which is defined as the sum of the squared distances from all points in a cluster domain to the cluster center (Tou & Gonzalez, 1974). First *K* random initial cluster centers are chosen. Then, each sample is assigned to a cluster based on the minimum distance to the cluster centers. Finally cluster centers are updated to the average of the values in each cluster. This is repeated until cluster centers no longer change. Steps in the K-Means algorithm are:

Step 1: Initialize *K* initial cluster centers randomly.

Step 2: For each pixel, calculate the distance to the cluster centers and assign the pixel to a cluster which has the minimum distance to its center.

Step 3: Calculate the average of the pixel values in each cluster and take them as new cluster centers.

Step 4: Repeat steps 2 and 3 until new cluster centers converge to the previous ones.

The K-means algorithm tends to find the local minima rather than the global minima. Therefore, it is heavily influenced by the choice of initial cluster centers and the distribution of data. Most of the time the results become more acceptable when initial cluster centers are chosen relatively far apart since the main clusters in a given data are usually distinguished in such a way. If the main clusters in a given data are too close to one another in the feature space, the K-means algorithm fails to recognize these clusters. For its improvement the K-means algorithm needs to be enhanced with the optimization technique in order to be less dependent on a given data set and initial cluster centers.

2.2 Simple Competitive Learning

Competitive learning model introduced by Rumelhart and Zipser in (Rumelhart & Zipser, 1986) is an interesting and powerful learning algorithm which can be used in unsupervised training for image classification (Hung, 1993). Several different competitive learning algorithms have been proposed by neural network researchers. These algorithms are capable of detecting various features represented in input signals. They have been applied in several different areas such as graph bipartitioning, vector quantization, etc (Hertz & Krogh, 1991). In this section the unsupervised simple competitive learning will be briefly presented.

The neural network models are characterized by the topology, activation function and learning rules. The topology of the simple competitive learning algorithm can be represented as a one-layered output neural net. Each input node is connected to each output

node. The number of input nodes is determined by the dimension of the training patterns. Unlike the output nodes in the Kohonen's feature map, there is no particular geometrical relationship between the output nodes in the simple competitive learning. In the following development, a 2-D one-layered output neural net will be used. During the process of training, the input vectors are fed into the network sequentially in time. The "trained" classes are represented by the output nodes and the center of each class is stored in the connection weights between input and output nodes.

The following algorithm outlines the operation of the simple competitive learning as applied to unsupervised training in (Hung, 1993); Let L denote the dimension of the input vectors, which for us is the number of spectral bands. We assume that a 2-D (N x N) output layer is defined for the algorithm, where N is chosen so that the expected number of the classes is less than or equal to N^2.

Step 1: Initialize weights $w_{ij}(t)$ (i = 1, …, L and j = 1, …, N x N) to small random values.

Steps 2 to 5 are repeated for each pixel in the training data set for each iteration.

Step 2: Present an input pixel X (t) = (x_1,…, x_L) at time t.

Step 3: Compute the distance d_j between x_i and each output node using

$$d_j = \sum_{i=1}^{L} (x_i - w_{ij}(t))^2 \tag{1}$$

where i, j, L, w_{ij} and x_i are similarly defined as in steps 1 and 2.

Step 4: Select an output node j^* which has the minimum distance. This node is called the best matching unit (BMU) or the winning node.

Step 5: Update weights of the winning node j^* using

$$w_{ij^*}(t+1) = w_{ij^*}(t) + \Delta(t)(x_i - w_{ij^*}(t)),\ i = 1, ..., L \text{ and } 1 \le j^* \le N \text{ x } N \tag{2}$$

where Δ(t) is a monotonically slowly decreasing function of t and its value is between 0 and 1.

Step 6: Select a subset of these N^2 output nodes as classes.

SCL shows stability over different run trials but this stable result is not always the global optima. In fact, in some cases the SCL converges to local optima over all run trials and the learning rate needs to be adjusted in the course of experimentation so that the global optimization can be achieved.

3. Swarm Intelligence

There are a number of techniques, developed for optimization, inspired by the behaviours of natural systems (Pham & Karaboga, 2000). In this study, we employ swarm intelligence, a natural optimization technique for optimizing both K-means and SCL algorithms.

3.1 Ant Colony Optimization

The ACO heuristic is inspired by the foraging behaviour of a real ant colony in finding the shortest path between the nest and the food. This is achieved by a deposited and accumulated chemical substance called *pheromone* by the passing ant which moves towards the food. In its searching the ant uses its own knowledge of where the smell of the food comes from (we call it as *heuristic information*) and the other ants' decision of the path toward

the food (*pheromone information*). After it decides its own path, it confirms the path by depositing its own pheromone making the pheromone trail denser and more probable to be chosen by other ants. This is a learning mechanism ants possess besides their own recognition of the path. As a result of t this consultation with the ants' behaviors already shown in searching for the food and returning to the nest, the best path which is the shortest is marked between the location of the nest and the location of the food.

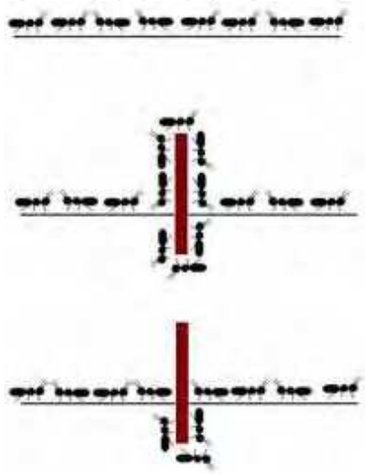


Figure 1. Ants finding the shortest path around an obstacle as a result of pheromone concentration

It was reported in the literature (Dorigo et al, 1996) that the experiments show when the ants have two or more fixed paths with the same length available from a nest to the food, they eventually concentrate on one of the paths and when the available paths are different in length they often concentrate on the shortest path. This is shown in Figure 1, when an obstacle is placed on the established path of ants, they first wander around the obstacle randomly. The ants going on a shorter path reach the food and return back to the nest more quickly. After a certain amount of time, the shorter path will be reinforced by pheromone. This path eventually becomes the preferred path of the ants (Zheng et al., 2003).

ACO uses this learning mechanism for the optimization. Furthermore, in the ACO algorithm, the pheromone level is updated based on the best solution obtained by a number of ants. The pheromone amount that is deposited by the succeeding ant is defined to be proportional to the quality of the solution it produces. For the real ants as shown in Figure 1, the best solution is the shortest path and it is marked with a strong pheromone trail. In the shortest path problem using the ACO algorithm, the pheromone amount deposited is inversely proportional to the length of the path.

In their research, Dorigo et al (1996) took the ant system as a colony of cooperating agents for solving the traveling salesman problem (TSP). Considering the short path problem, suppose for any pair of nodes V_i and V_j on the graph G, there is a connection cost attached to the edge (V_i, V_j) and the pheromone trail and heuristic information are stored on the edge.

The goal of an ACO heuristic is then to find the shortest path on graph G. In ACO heuristic, m artificial ants are normally used to find the best solution. Suppose an ant k is in vertex V_i at certain step i during its search process. This ant selects the connection with the probability (Dorigo et al., 1996):

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{h\in allowed_{k}(t,I)}\tau_{ih}^{\alpha}\eta_{ih}^{\beta}}, & j \in allowed_{k}(t,I) \\ 0, & otherwise \end{cases} \tag{3}$$

where P^{k}_{ij} is the probability of ant k choosing the path (V_i, V_j) from V_i. Parameters τ_{ij} and η_{ij} are the pheromone and heuristic information assigned to the edge (V_i, V_j) respectively, α and β are constants that determine the relative influence of the pheromone and heuristic information, and $allowed_k(t, I)$ is the set of vertices which is allowed to be visited according to problem constraints.

Then ant k moves and deposits a pheromone on the trail, which is defined below:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, & if\ ant\ k\ uses\ connection\ (V_i, V_j)\ in\ its\ solution \\ 0, & otherwise \end{cases} \tag{4}$$

where Q is a positive constant and L_k is the cost of the path used by ant k. After all m ants completed their path finding, the pheromone on each edge is updated according to the following formula:

$$(1-\rho)\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \Rightarrow \tau_{ij} \tag{5}$$

where ρ is the evaporation factor ($0 \le \rho \le 1$) which causes the earlier pheromones vanish over the iterations. Therefore, as the solution becomes better, the corresponding pheromone have more effect on the next solution rather than the earlier pheromones which correspond to the initial undesired solutions found.

This pheromone information will be a guide for the new set of ants. Each time, the current best solution is saved, and this process will be repeated until a termination criterion is met.

3.2 Particle Swarm Optimization

The PSO algorithm is inspired by the group behavior of schools of fish, flocks of birds and swarms of insects. As an example, birds are likely to find food in flocks, without knowing its location in advance. It seems that members of the flock buildup their intuition in order to find their nutriment. As sociobiologist E. O. Wilson (Wilson, 1975) has written, in reference to fish schooling, "In theory at least, individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpredictably distributed in patches." (p. 209)

The PSO algorithm consists of a swarm of particles flying through the search space (Kaewkamnerdpong & Bentley, 2005). Each particle's position is a potential solution to the problem. Each particle's velocity is modified based on its distance from its personal best

position and the global best position. In other words the particles move according to their experience and that of their neighbors which yields to the best fitness value.
Each particle *i* maintains the following information (van der Merwe & Engelbrecht, 2003):

- x_i, the current position of the particle,
- v_i, the current velocity of the particle,
- y_i, the personal best position of the particle (*pbest*); the best position visited so far by the particle, and
- $\hat{y}$, the global best position of the swarm (*gbest*); the best position visited so far by the entire swarm.

The objective function evaluates the positions of the particles. Personal best position (*pbest*) is then obtained as follows (van der Merwe & Engelbrecht, 2003):

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \tag{6}$$

where *f* is the objective function. The global best position (*gbest*) is obtained as follows (van der Merwe & Engelbrecht, 2003):

$$\hat{y}(t) \in \{y_0, y_1, ..., y_s\} = \min\{f(y_0(t)), f(y_1(t)), ..., f(y_s(t))\} \tag{7}$$

For each iteration of a PSO algorithm, v_i and x_i are updated as follows (van der Merwe & Engelbrecht, 2003):

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(y_i(t) - x_i) + c_2 r_2(t)(\hat{y}(t) - x_i(t)) \tag{8}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{9}$$

where ω is the inertia weight which serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity. The cognitive component, $y_i(t)$–x_i represents the particle's own experience as to where the best solution is. The social component, $\hat{y}(t) - x_i$ represents the belief of the entire swarm as to where the best solution is. c_1 and c_2 are acceleration constants and $r_1(t)$, $r_2(t) \sim U(0,1)$,where $U(0,1)$ is a random number between 0 and 1.
The PSO algorithm is repeated until a termination criterion is reached or the changes in velocity get near to zero. A fitness function is used to evaluate the optimality of the solution.
The following algorithm outlines a PSO based image classification (Omran et al., 2002). In this algorithm, a single particle x_i represents N cluster means such that $x_i=(m_{i1},...,m_{ij},\ldots,m_{iN})$ where m_{ij} represents the *j*-th cluster centroid vector of the *i*-th particle. Therefore, a swarm represents a number of candidate cluster centers. The fitness of each set of cluster is measured using:

$$f(x_i, Z_i) = \omega_1 \bar{d}_{\max}(Z_i, x_i) + \omega_2(z_{\max} - d_{\min}(x_i)) \tag{10}$$

where $z_{max}=2s$ -1 for an s-bit image; *Z* is a matrix representing the assignment of pixels to clusters of particle *i*. Each element z_{ijp} indicates if pixel z_p belongs to cluster C_{ij} of particle *i*. The constants w_1 and w_2 are user defined constants. Also,

$$\bar{d}_{\max}(Z_i, x_i) = \max_{j=1,\ldots,N_c} \{ \sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij}) / | C_{ij} | \} \tag{11}$$

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

$$d_{\min}(x_i) = \min_{\forall j_1, j_2, j_1 \neq j_2} \{ d(m_{ij_1}, m_{ij_2}) \} \tag{12}$$

is the minimum Euclidean distance between any pair of clusters. The algorithm is as follows:

Step 1: Initialize cluster centers for each particle randomly.

Step 2: For each particle, assign each pixel to a cluster that has the minimum distance to its cluster center.

Step 3: Calculate the fitness function for each particle and find the global best solution.

Step 4: Update the cluster centers using Eqs. (8) and (9).

Step 5: Repeat the procedure until the stopping criterion is reached.

4. Swarm Intelligence and K-means

4.1 The Hybrid ACO-K-means Algorithm

We propose a hybrid ACO-K-means algorithm which uses the ACO to improve the performance of the K-means algorithm for clustering. The proposed algorithm starts by choosing the number of clusters and a random initial cluster center for each cluster. ACO plays its part in assigning each pixel to a cluster. This is done according to a probability which is inversely proportional to the distance (similarity) between the pixel and cluster centers and a variable, τ, representing the pheromone level. We define pheromone to be proportional to minimum distance between each pair of cluster centers and inversely proportional to the distances between each pixel and its cluster center. So the pheromone gets larger when the cluster centers are far apart and clusters are more compact (our criterion for best solution), making the probability of assigning a pixel to that cluster high. Pheromone evaporation is considered to weaken the influence of the previously chosen solutions, which are less likely to be desired. Similar to the K-means algorithm, at this stage new cluster centers are updated by calculating the average of the pixels in each cluster and this will be repeated until cluster centers no longer change. But unlike K-means, this algorithm doesn't stop here. We assume that the clustering job is performed by an ant and there are *m* ants repeating this job, each with their own random initialization, and they all will end up with a solution. A criterion is defined to find the best solution and the pheromone level is updated accordingly for the next set of *m* ants as a leading guide. If the termination criterion is satisfied, the algorithm will be terminated. Hence, an optimal solution is obtained.

The algorithm starts by assigning a pheromone level τ and a heuristic information η to each pixel. Then each ant will assign each pixel to a cluster with the probability distribution *P* derived from Eq. (13), (Dorigo et al., 1996):

$$P_i(X_n) = \frac{[\tau_i(X_n)]^\alpha [\eta_i(X_n)]^\beta}{\sum_{j=0}^{K} [\tau_j(X_n)]^\alpha [\eta_j(X_n)]^\beta} \tag{13}$$

where $P_i(X_n)$ is the probability of assigning pixel X_n to cluster i, $\tau_i(X_n)$ and $\eta_i(X_n)$ are the pheromone and heuristic information assigned to pixel X_n in cluster i respectively, α and β are constant parameters that determines the relative influence of the pheromone and heuristic information, and K is the number of clusters. Heuristic information $\eta_i(X_n)$ is obtained from:

$$\eta_i(X_n) = \frac{\kappa}{CDist(X_n, CC_i) * PDist(X_n, PC_i)} \tag{14}$$

where X_n is the n^{th} pixel, CC_i is the i^{th} spectral cluster center and PC_i is the i^{th} spatial cluster center. *CDist* (X_n, CC_i) is the Euclidean distance between X_n and CC_i , considering the color features of the pixels and *PDist* (X_n, PC_i) is the Euclidean distance between X_n and PC_i, considering the position of the pixels on the image. Constant κ is used to balance the value of η with τ.

The value for the pheromone level τ assigned to each pixel is initialized to 1 so that it does not have any effect on the probability at the beginning. This pheromone gets larger over the iterations which we describe later.

Suppose m number of ants is chosen for clustering an image. Each ant is giving its own clustering solution. After m ants have done their clustering, the current best solution is chosen and the assigned pheromone to this solution is incremented. All the cluster centers are updated using the cluster centers of the current best solution. The next set of ants is inspired from the previous set. In each of the iterations, each one of the m ants search for a solution based on its own heuristic knowledge and the best solution found by the previous m ants. This is repeated a certain amount of times until the overall best solution is obtained.

The best solution of the m solutions found in each of iterations is selected according to two factors; Euclidean distance between cluster centers in terms of spectral values (separateness of clusters), and sum of the Euclidean distances between each pixel and its cluster center, in terms of spectral and spatial values (similarity and compactness of each cluster). To choose the best solution: 1) the Euclidean distance between cluster centers in terms of spectral signatures should be large so the clusters are more distinct, 2) the sum of the Euclidean distances between each pixel and its cluster center in terms of spectral signatures should be small so that each cluster becomes more homogeneous, and 3) the sum of the Euclidean distances between each pixel and its cluster center in terms of spatial signatures should be small so that each cluster becomes more compact. To achieve the first one, for each clustering performed by ant k (k = 1,…,m), we compute the distances between every pair of cluster centers and sort these distances. Then we select the minimum distance $Min(k)$. Now we compare all these minimums performed by all the ants, and select the maximum of them [$MinMax(k)$]. To achieve the second and third, for each clustering performed by ant k we compute the sum of the distances between each pixel and its cluster center, and sort these sums of the distances. Then we select the maximum and compare all these maximums performed by all ants, and select the minimum of them. The second maximum and third maximum of the solutions are compared in the same way and the minimum is selected. Each time a solution is selected it gets an additional vote. The solution with the largest vote is selected as the best solution.

After the best solution is found, the pheromone value is updated according to Eq. (15) (Li & Xu, 2003):

$$\tau_i(X_n) \leftarrow (1- \rho)\, \tau_i(X_n) + \Sigma_i \Delta\tau_i(X_n) \tag{15}$$

where ρ is the evaporation factor ($0 \le \rho \le 1$) which causes the earlier pheromones vanish over the iterations. Therefore as the solution becomes better, the corresponding pheromone have more effect on the next solution rather than the earlier pheromones which correspond to the initial undesired solutions found. The parameter $\Delta\tau_i(X_n)$ in Eq. (15) is the amount of pheromone added to previous pheromone by the succeeded ant, which is obtained from:

$$\Delta\tau_i(X_n) = \begin{cases} \dfrac{Q * Min(k')}{AvgCDist(k',i) * AvgPDist(k',i)} & \text{if } X_n \text{ is a member of cluster i.} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

In Eq. (16), Q is a positive constant which is related to the quantity of the added pheromone by ants. $Min(k')$ is the minimum distance between every two cluster centers obtained by ant k' (the winner ant) in spectral feature space. $AvgCDist(k',i)$ is the average of the spectral Euclidean distances and $AvgPDist(k',i)$ is the average of the spatial Euclidean distances between all pixels in cluster i and their spectral cluster center and spatial cluster center obtained by ant k', respectively. $Min(k')$ causes the pheromone become larger when clusters get more far apart and hence raise the probability. $AvgCDist(k',i)$ and $AvgPDist(k',i)$ cause the pheromone become larger when the cluster has more similar pixels and is more compact respectively. In other words, the more the *Min(k')* is, the more far apart our clusters are which is desired and the larger the pheromone is. The less the *AvgCDist(k',i)* and *AvgPDist (k',i)* are, the more similar and compact our clusters are which is desired and the larger the pheromone is.

Next, cluster centers are updated using the cluster centers of the best solution. This algorithm is repeated a certain amount of times until the very best solution is obtained.

The Hybrid ACO-K-means algorithm is described below:

Step 1: Initialize pheromone level assigned to each pixel to 1, the number of clusters to K and number of ants to m.

Step 2: Initialize m sets of K random cluster centers to be used by m ants.

Step 3: Let each ant, assign each pixel X_n to one of the clusters (i), randomly, with the probability distribution $P_i(X_n)$ given in Eq. (13).

Step 4: Calculate new cluster centers; If the new cluster centers are approximately equal to the old ones, go to next step. Otherwise, go to Step 3.

Step 5: Save the best solution among the m solutions found.

Step 6 Update the pheromone level for each pixel according to the best solution using Eqs. (15) and (16).

Step 7: Assign cluster center values of the best clustering solution to the clusters centers of all ants.

Step 8: If the termination criterion is satisfied go to the next step. Otherwise, go to Step 3.

Step 9: Output the optimal solution.

4.2 The Hybrid PSO-K-means Algorithm

As discussed in section 3.2, in the PSO based clustering presented in (Omran et al., 2002) the cluster centers assigned to particles were initialized randomly. Each pixel was distributed to a cluster with minimal Euclidean distance. Then PSO was used to refine the cluster centers using a fitness function. In (van der Merwe & Engelbrecht, 2003) the K-means algorithm was

applied to feed one particle of the initial swarm and the rest of the swarm were initialized randomly. Then the same algorithm described above is employed. We propose a new hybridization of PSO-K-means algorithm where the K-means is applied to all particles and solutions are evaluated in a way similar to the evaluation used in the proposed ACO-K-means algorithm.

The proposed PSO-K-means algorithm is presented as follows:

Step 1: Initialize the number of clusters to K and number of particles to m.

Step 2: Initialize m sets of K random cluster centers to be used by m particles.

Step 3: For each particle, let each pixel x belong to a cluster in which it has the smallest Euclidean distance to the centroid.

Step 4: Calculate new cluster centers; If the new cluster centers converge to the old ones, go to the next step. Otherwise, go to Step 3.

Step 5: Save the best solution found so far performed by each particle. Call it *pbest* or personal best solution.

Step 6: Save the best solution among the m personal best solutions found. Call it *gbest* or global best solution.

Step 7: Update cluster centers of each particle according to the cluster center values of the *pbest* and *gbest* solution, using Eqs. (8) and (9).

Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.

Step 9: Output the optimal solution.

5. Swarm Intelligence and Simple Competitive Learning

5.1 The Hybrid ACO-SCL Algorithm

We apply the ACO to simple competitive learning algorithm and investigate its performance. Similar to our previous hybrid algorithms described in section 4, the pheromone and heuristic information are defined to satisfy the clustering criteria which include the similarity of data in each cluster, distinction of the clusters and compactness of each cluster. At the end of all iterations, the best solution is selected in the same way as what we used for the ACO-K-means algorithm.

The ACO-SCL algorithm is described as follows. Let L denote the dimension of the input vectors, which for us is the number of spectral bands. We assume that a 2-D ($N \times N$) output layer is defined for the algorithm, where N is chosen so that the expected number of the classes is less than or equal to N^2. Here weights of the nodes contain cluster center values.

Step 1: Initialize the number of clusters to K and the number of ants to m. Initialize pheromone level assigned to each pixel to 1 so that it does not have any effect on the probability using later at the beginning.

Step 2: Initialize m sets of K different random cluster centers to be used by m ants.

Step 3: Let each ant, assign each pixel X_n to one of the clusters (i), randomly, with the probability distribution $P_i(X_n)$ given in Eq. (13) where heuristic information $\eta_i(X_n)$ is obtained from Eq. (14).

Step 4: For each input pixel the cluster center of the cluster which it belongs to is considered as the BMU. Calculate new cluster centers using:

$$C_i(t+1) \leftarrow C_i(t) + \Delta(t)(x_i - C_i(t)),\ i = 1, ..., L \tag{17}$$

where Δ(t) is a monotonically slowly decreasing function of t and its value is between 0 and 1. *L* is the number of spectral bands.
Step 5: Save the best solution among the *m* solutions found.
Step 6: Update the pheromone level for each pixel according to the best solution. The pheromone value is updated according to Eq. (15).
Step 7: Assign cluster center values of the best clustering solution to the clusters centers of all ants.
Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step3.
Step 9: Output the optimal solution.

5.2 The Hybrid PSO-SCL Algorithm

The PSO-SCL algorithm which combines the PSO and SCL, is described as follows. Let *L* denote the dimension of the input vectors, which for us is the number of spectral bands. We assume that a 2-D ($N \times N$) output layer is defined for the algorithm, where *N* is chosen so that the expected number of the classes is less than or equal to N^2. Here weights of the nodes contain cluster center values.
Step 1: Initialize the number of clusters to *K* and the number of particles to *m*.
Step 2: Initialize *m* sets of *K* different random cluster centers to be used by *m* particles.
Step 3: For each particle, let each pixel *x* belong to a cluster in which it has the smallest Euclidean distance to the centroid.
Step 4: For each input pixel the cluster center of the cluster which it belongs to is considered as the BMU. Calculate new cluster centers using Eq. (17).
Step 5: Save the best solution found so far performed by each particle. Call it *pbest* or personal best solution.
Step 6: Save the best solution among the *m* personal best solutions found. Call it *gbest* or global best solution.
Step 7: Update cluster centers of each particle according to the cluster center values of the *pbest* and *gbest* solution, using Eq. (8).
Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.
Step 9: Output the optimal solution.

6. Simulation Results

Experimental results from our proposed hybrid algorithms were compared with those of the K-means and the SCL algorithms, and discussed in this section. Since the SCL is very dependent on the learning rate, i.e. $\Delta(t)$ in Eq. (17), we performed some experiments on choosing a value for $\Delta(t)$. Considering that $\Delta(t)$ is a monotonically slowly decreasing function of *t* and its value is between 0 and 1, we suggest the following formula:

$$\Delta(t) = \frac{0.2}{t^r + 1} \tag{18}$$

where *t* and *r* denote iteration and a rate which is a constant that we obtained by experiments, respectively. The experiments were performed over 20 run trials on several different images, for *r* from 10 to 50 incrementing by 10. Two of them reported in this chapter. Experiments showed better results for $r = 10$. Therefore, the experiment was

repeated similarly but this time for r from 1 to 10 incrementing by 1. This experiment showed better results for r between 1 and 5. In our experiments r is chosen to be 2.

The ACO-K-means and ACO-SCL algorithms were shown to be dependant on parameterization as well. Parameters used in these algorithms, other than r include κ, Q, ρ, α, and β. Parameters α, β and κ are used to keep the values of τ and η in the same order. Parameter Q controls the added amount of pheromone and ρ eliminates the influence of the earlier added pheromone. Considering that r should be between 1 and 5 from the previous experiment, r was chosen to be 2. Evaporation factor was set to be $\rho = 0.8$. According to the performance of the experiments, parameters κ and Q were shown to have little influence on the results, while α and β were more influential. The parameter values tested were as follows: κ =1000 and 10000, Q =10 and 100, α =0.1 to 50 incrementing by 10, and β =0.1 to 50 incrementing by 10. Each experiment was performed for 20 run trials on each image. There were unacceptable results when β =0.1. There were good results when α =0.1, for images shown here but they were unstable. There were some sets of parameters that still did well for one of the images but not for the other. Knowing that α should be small while β should not be small, we set up another experiment: κ =1000 and 10000, Q =10 and 100, α =0.1 to 2 incrementing by 0.1, β =50 to 5 decrementing by 5. All the results were acceptable but not all of them were stable. So in this experiment stability of the results was examined. Experimental results show that β should not be very large, otherwise it becomes unstable. When β is chosen to be 5 and α is between 0.1 and 2, the result showed to be more stabled. From these sets of experiments, the chosen parameters are as follows: $r = 2$, $\rho = 0.8$, $\alpha = 2$, $\beta = 5$, $\kappa = 1000$, and $Q = 10$. The number of ants was chosen to be $m = 5$.

The PSO-K-means and PSO-SCL algorithms also include a set of parameters to be determined empirically. The parameters were chosen as suggested by (van der Merwe and Engelbrecht, 2003) which resulted in good convergence. Parameters were set as follows: c_1 , $c_2 = 1.49$ and $\omega = 0.72$. The number of ants was chosen to be $m = 10$.

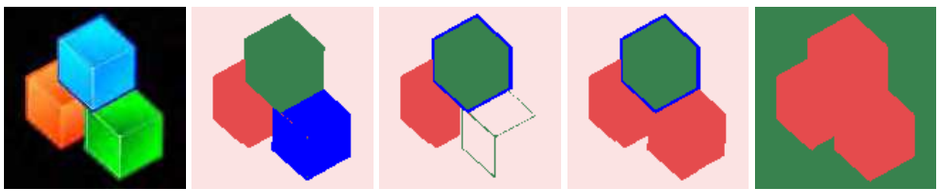
We examined the proposed hybrid algorithms and compared the results with those of the K-means and the SCL algorithms in Figs. 2 to 5. Images used include flamingo, cubes, aurora and river. The number of clusters to be found in all images is 3 except for cubes which is 4. The most dominant results of the algorithms over 20 different run trials are presented. The improvement of the ACO and PSO on the K-means algorithm is obvious in all of the images tested. In cubes, flamingo and Aurora images it can be seen that the K-means algorithm has unstable results and in some cases it misses some clusters while the ACO-K-means and PSO-K-means algorithms are more stable and they clearly recognize the clusters. In the river image the results show that the K-means algorithm can generate stable results and the ACO-K-means algorithm seems to be less stable, but it is apparent that even for this image, the ACO-K-means algorithm can improve the classification results. This is also the case with the ACO-SCL and PSO-SCL algorithms as opposed to SCL in Aurora image. Results on Aurora image clearly show that the ACO and PSO algorithms can improve the SCL in cases where the SCL algorithm is trapped into local optima. To further investigate the behavior of the algorithms described, we obtained the classification accuracy percentage of the results on the river image. Each algorithm is run 30 times on the river image, shown in Fig. 5 (a). Then , by comparing the classification results with the ground truth data, shown in Fig. 5 (h), the error matrix for each classified image is calculated. The best, worst and average cases are shown in Fig. 6. The stability of the SCL algorithm over the ACO-SCL and the PSO-SCL

algorithms can be inferred from this figure. But, as it was stated before in the aurora image this stable result is not always a global optima. Similarly, it can be inferred from Fig. 6 that the K-means algorithm is more stable than the ACO-K-means algorithm in the case of river image. Nevertheless, results of the ACO-K-means algorithm include some very good results with much higher classification accuracy percentage than those of the K-means algorithm.

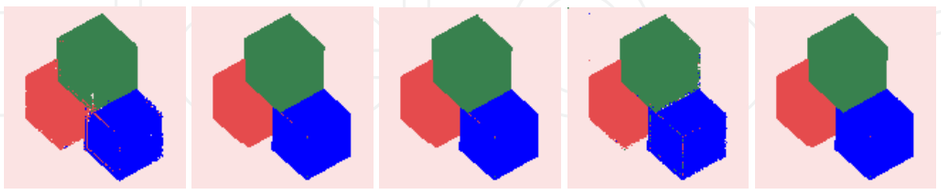
(a) (b_1) (b_2) (c) (d)

(e) (f) (g_1) (g_2) (g_3)

Figure 2. The most dominant classified results among 20 runs (a) Original image, (b1 & b2) K-means (c) Hybrid ACO-K-means, (d) Hybrid PSO-K-means, (e) SCL, (f) Hybrid ACO-SCL, (g1, g2, & g3) Hybrid PSO-SCL

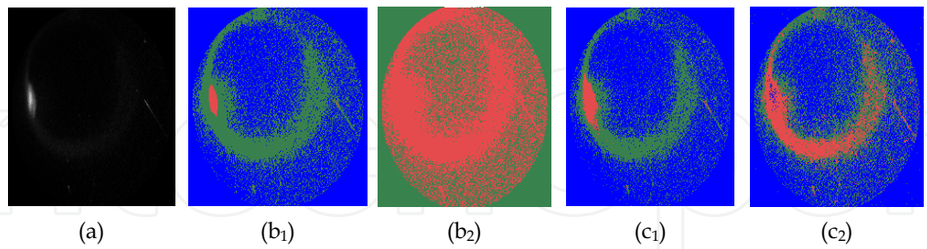


(a) (b_1) (b_2) (b_3) (b_4)

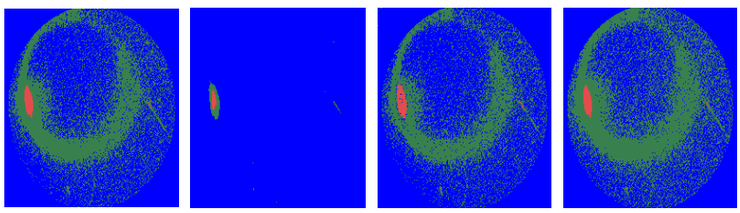


(c) (d) (e) (f) (g)

Figure 3. The most dominant classified results among 20 runs (a) Original image, (b_1, b_2, b_3 & b_4) K-means, (c) Hybrid ACO-K-means, (d) Hybrid PSO-K-means, (e) SCL, (f) Hybrid ACO-SCL, (g) Hybrid PSO-SCL

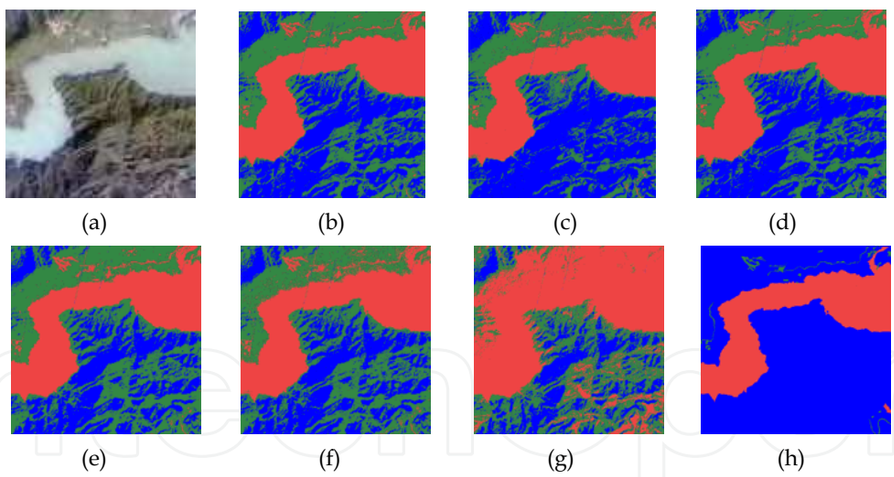


(a) (b_1) (b_2) (c_1) (c_2)



(d) (e) (f) (g)

Figure 4. The most dominant classified results among 20 runs (a) Original image, (b_1 & b_2) K-means, (c_1 & c_2) Hybrid ACO-K-means, (d) Hybrid PSO-K-means, (e) SCL, (f) Hybrid ACO-SCL, (g) Hybrid PSO-SCL



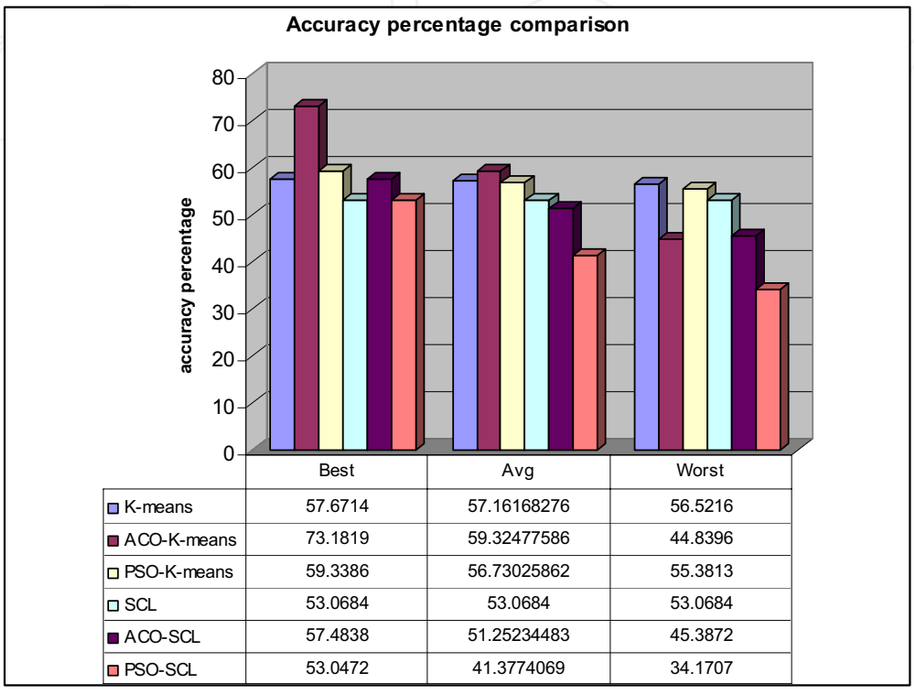
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5. The most dominant classified results among 20 runs (a) Original image, (b) K-means, (c) Hybrid ACO-K-means, (d) Hybrid PSO-K-means, (e) SCL, (f) Hybrid ACO-SCL, (g) Hybrid PSO-SCL, (h) The ground truth data

Besides, the stability of the K-means algorithm over the ACO-K-means algorithm inferred from fig. 6, is a particular case, i.e. for the river image. The K-means algorithm is not stable

in general. In fact in the case of flamingo and cubes, ACO-K-means algorithm produced more stabled results compared to K-means algorithm.



	Best	Avg	Worst
K-means	57.6714	57.16168276	56.5216
ACO-K-means	73.1819	59.32477586	44.8396
PSO-K-means	59.3386	56.73025862	55.3813
SCL	53.0684	53.0684	53.0684
ACO-SCL	57.4838	51.25234483	45.3872
PSO-SCL	53.0472	41.3774069	34.1707

Figure 6. Comparison of the results generated by algorithms using error matrix evaluation on the river image

7. Conclusion

Experimental results showed that SI techniques can improve the K-means and the SCL algorithms in recognizing the clusters. The K-means algorithm often fails to realize clusters since it is heavily dependent on the initial cluster centers. The ACO-K-means and PSO-K-means algorithms provides a larger search space compared to the K-means algorithm. By employing these algorithms for clustering, the influence of the improperly chosen initial cluster centers will be diminished over a number of iterations. Therefore, these algorithms are less dependent on randomly chosen initial seeds and is more likely to find the global optimal solution.

We have also shown that SI can be beneficial to the SCL algorithm. SI can help SCL find the global optima using the same parameter set and learning rate as those used in the SCL and recognize the clusters where the SCL fails to do, in some cases. This can be advantageous since for SCL to find the global optima the learning rate should be adjusted in the course of experimentation.

8. References

Dorigo, M.; Maniezzo, V. & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents, In: *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol. 26, 1996, pp. 29-41

Gonzalez, R.C. & Woods, R.E. (1992). Digital Image Processing. Addison-Wesley, 1992

Hertz, J.; Krogh, A. & Palmer R.G. (1991). Introduction to the theory of neural computation, Addison-Wesley, Redwood City, 1991

Hung C.C. (1993). Competitive learning networks for unsupervised training, *INT. J. Remote Sensing*, vol. 14, no. 12, 1993, pp. 2411-2415

Kaewkamnerdpong, B. & Bentley, P.J. (2005). Perceptive Particle Swarm Optimisation: an Investigation, *IEEE Swarm Intelligence Symposium*, pp. 169-176, Pasadena, CA, USA, June, 2005

Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization, In: *Proceedings of IEEE International Conference on Neural Networks*, pp.1942-1948, 1995

Li, Y. & Xu, Z. (2003). An Ant Colony Optimization Heuristic for Solving Maximum Independent Set Problems, In: *Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, pp. 206-211, Xi'an, China, Sept. 2003

McQueen, J.B. (1967). Some Methods of Classification and Analysis of Multivariate Observations, In L. M. LeCam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability*, pp. 281-297, University of California Press, Berkley, CA, 1967

Omran, M.; Salman, A. & Engelbrecht, A.P. (2002). Image Classification Using Particle Swarm Optimization, 2002

Pham, D.T. & Karaboga, D. (2000). Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search*, Simulated Annealing and Neural Networks*, Springer, 2000

Rumelhart, D.E. & Zipser, D. (1986). Feature discovery by competitive learning, In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds. J.L. McClelland & D.E. Rumelhart, MIT Press, Cambridge, M.A., 1986, pp. 151-193

Saatchi, S. & Hung, C.C. (2005). Hybridization of the Ant Colony Optimization with the K-means Algorithm for Clustering, Image Analysis, *Proceedings Lecture Notes in Computer Science* 3540, 2005, pp. 511- 520

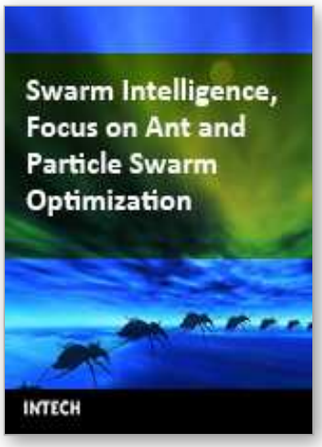
Saatchi, S.; Hung, C.C. & Kuo, B.C. (2006). A comparison of the improvement of k-means and simple competitive learning algorithms using ant colony optimization, In *Proceedings of the 7th International Conference on Intelligent Technology*, Taipei, Taiwan, December, 2006

Tou, J.T. & Gonzalez, R.C. (1974). Pattern Recognition Principles, Addison-Wesley, 1974

van der Merwe, D.W. & Engelbrecht A.P. (2003). Data Clustering Using Particle Swarm Optimization, 2003 *Congress on Evolutionary Computation*, 2003, pt. 1, Vol.1, pp. 215-220

Wilson, E.O. (1975). Sociobiology: the new synthesis, Belk Press, Cambridge, MA, 1975

Zheng, H.; Zheng, Z. & Xiang Y. (2003). The application of ant colony system to image textute classification [textute read texture], In: *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1491-1495, Xi'an, China, Nov. 2003



Swarm Intelligence, Focus on Ant and Particle Swarm Optimization
Edited by FelixT.S.Chan and Manoj KumarTiwari

ISBN 978-3-902613-09-7
Hard cover, 532 pages
Publisher I-Tech Education and Publishing
Published online 01, December, 2007
Published in print edition December, 2007

In the era globalisation the emerging technologies are governing engineering industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from the nature and implant it in the engineering sciences. This way of thinking led to emergence of many biologically inspired algorithms that have proven to be efficient in handling the computationally complex problems with competence such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms the present book on "Swarm Intelligence: Focus on Ant and Particle Swarm Optimization" aims to present recent developments and applications concerning optimization with swarm intelligence techniques. The papers selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. In addition to the introduction of new concepts of swarm intelligence, this book also presented some selected representative case studies covering power plant maintenance scheduling; geotechnical engineering; design and machining tolerances; layout problems; manufacturing process plan; job-shop scheduling; structural design; environmental dispatching problems; wireless communication; water distribution systems; multi-plant supply chain; fault diagnosis of airplane engines; and process scheduling. I believe these 27 chapters presented in this book adequately reflect these topics.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sara Saatchi and Chih-Cheng Hung (2007). Swarm Intelligence and Image Segmentation, Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, FelixT.S.Chan and Manoj KumarTiwari (Ed.), ISBN: 978-3-902613-09-7, InTech, Available from: http://www.intechopen.com/books/swarm_intelligence_focus_on_ant_and_particle_swarm_optimization/swarm_intelligence_and_image_segmentation

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen