



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Hybrid Evolutionary Algorithm Based on ACO and SA for Cluster Analysis

¹T. Niknam, ²J. Olamaei and ³B. Amiri

¹Department of Electronic and Electrical, Shiraz University of Technology,
Modars Blvd. Shiraz, Iran

²Islamic Azad University, South Tehran-Branch, Tehran, Iran

³Department of Information Technology and Computer Engineering,
Fars Science and Research Branch of Islamic Azad University, Shiraz, Iran

Abstract: This study presents an efficient hybrid evolutionary optimization algorithm based on combining Ant Colony Optimization (ACO) and Simulated Annealing (SA), called ACO-SA, for optimal clustering N object into K clusters. The new ACO-SA algorithm is tested on several data sets and its performance is compared with those of ACO, SA and K-means clustering. The simulation results show that the proposed evolutionary optimization algorithm is robust and suitable for data clustering.

Key words: Ant colony optimization (ACO), K-means clustering, data clustering, hybrid evolutionary optimization algorithm, simulated annealing (SA)

INTRODUCTION

Data mining uses a variety of data analysis tools to discover patterns and relationships of physical variables in different data sets. This technique is used in multidisciplinary fields. Clustering is one of many powerful data mining techniques that can discover intentional structures in data (Kao *et al.*, 2008). It groups instances which have similar features and builds a concept hierarchy. As a result, it often extracts new knowledge from a database. Because of the ability to find the intentional descriptions, it is one of the very important techniques in managing databases (Lu *et al.*, 2008). Clustering problems arise in many different applications such as knowledge discovery, data compression and vector quantization and pattern recognition and classification. There are many methods in clustering. The existing methods can be roughly divided into hierarchical, iterative and optimization methods. Iterative methods are usually based on the K-means algorithm, which applies two optimization steps iteratively (Amir and Lipika, 2007): (i) calculate optimal partitions for the given codebook and (ii) calculate a new codebook as the cluster centroids. K-means has two shortcomings: dependency on the initial state and convergence to local optima and also global solutions of large problems cannot find with reasonable amount of computation effort (Fathian *et al.*, 2007). In order to overcome local optima problem lots of studies have done in clustering. In these studies, researchers

have used meta-heuristic algorithms like genetic algorithm, simulated annealing and tabu search. Over the last decade, modeling the behavior of social insects, such as ants and bees, for the purpose of search and problem solving has been the context of the emerging area of swarm intelligence.

The ant colony optimization (ACO) paradigm was first proposed by Dorigo (2006). Since then, a number of studies based on ant colony algorithm have been carried out in order to solve some optimization problems such as the traveling salesman problem, unit commitment, etc. (Ho *et al.*, 2006; Huang, 2001; Merkle *et al.*, 2002; Kwang and Sun, 2003); however, in these studies, the search domains are in discrete form. In addition, a number of methods based on ant colony have been presented to solve the optimization problem in both continuous and discrete search domains such as economic dispatch, reactive power pricing in restructured networks (Niknam *et al.*, 2005a, b). The main problem associated with these methods is that the time of convergence increases drastically when the number of variables is increased. In this study, a new hybrid algorithm based on the ant colony algorithm and simulated annealing is presented to solve optimization problems in continuous and discrete domains simultaneously. Since the SA is a good local search algorithm, in the proposed hybrid algorithm we use SA as a local search in ACO. The proposed hybrid algorithm is applied for optimal clustering N object into K clusters. The simulation results

show that this algorithm not only has a better response but also converges more quickly than ordinary evolutionary methods like ACO, SA and K-means.

CLUSTER ANALYSIS PROBLEM

Cluster analysis is an effective tool in scientific or managerial inquiry. The k-means clustering method is one of the simplest unsupervised learning algorithms for solving the well-known clustering problem. The goal is to divide the data points in a data set into K clusters fixed a priori such that some metric relative to the centroids of the clusters (called the fitness function) is minimized. The algorithm consists of two stages: an initial stage and an iterative stage. The initial stage involves defining K initial centroids, one for each cluster. These centroids must be selected carefully because of differing initial centroids causes differing results. One policy for selecting the initial centroids is to place them as far as possible from each other. The second, iterative stage repeats the assignment of each data point to the nearest centroid and K new centroids are recalculated according to the new assignments. This iteration stops when a certain criterion is met, for example, when there is no further change in the assignment of the data points. Given a set of n data samples, suppose that we want to classify the data into K groups, the algorithm aims to minimize a fitness function, such as a squared error function defined as:

$$F = \sum_{j=1}^K \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \tag{1}$$

where, $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between the i^{th} data point of the data sample $x_i^{(j)}$ (which was classified into the j^{th} group) and the j^{th} cluster center c_j , $1 \leq i \leq n$, $1 \leq j \leq K$ and is an indicator of the distance of the n data samples from their respective cluster centroids. The k-means clustering algorithm is summarized in the following steps (Fathian *et al.*, 2007):

Step 1: Place K points into the space represented by the objects that are being clustered. These points represent the initial group centroids.

Step 2: Assign each object to the group that has the closest centroid.

Step 3: When all objects have been assigned, recalculate the positions of the K centroids.

Step 4: Repeat steps 2 and 3 until a certain criterion is met, such as the centroids no longer moving or a preset

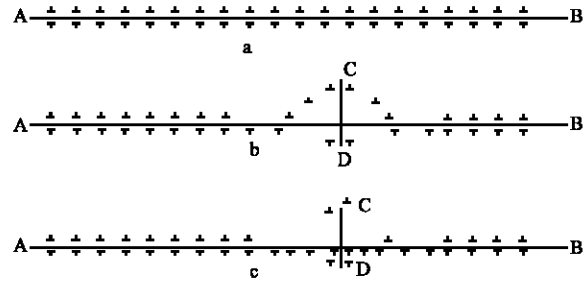


Fig. 1: An example of finding the shortest path by ants

number of iterations have been performed. This results in the separation of objects into groups for which the score of the fitness function is minimized.

Ant colony algorithm: Ants are insects which live together. Since they are blind animals, they find the shortest path from their nest to food with the aid of pheromone. Pheromone is the chemical material deposited by the ants, which serves as critical communication medium among ants, thereby guiding the determination of the next movement. On the other hand, ants find the shortest path based on intensity of pheromone deposited on different paths 15-23. Assume that ants want to move from point A to B (Fig. 1).

At first, if there is no obstacle, all of them will move along the straight path (AB) (Fig. 1a). At the next stage, assume that there is an obstacle; in this case, ants will not be able to follow the original trail in their movement. Therefore, randomly, they turn to the left (ACB) or to the right (ADB) (Fig. 1b). Since ADB path is shorter than ACB, the intensity of pheromone deposited on ADB is more than the other path. So ants will be increasingly guided to move on the shorter path (Fig. 1c). This behavior forms the fundamental paradigm of the ant colony system.

As was indicated in Fig. 1, the intensity of deposited pheromone is one of the most important factors for ants to find the shortest path. Generally, the intensity of pheromone and path length are two important factors that should be used to simulate the ant system. To select the next path, the state transition probability is defined as follows:

$$P_{ij} = \frac{(\tau_{ij})^{\gamma_1} (1/L_{ij})^{\gamma_2}}{\sum_{j=1}^N (\tau_{ij})^{\gamma_1} (1/L_{ij})^{\gamma_2}} \tag{2}$$

where, τ_{ij} and L_{ij} are the intensity of pheromone and the length of path between nodes j and i, respectively. γ_1 and γ_2 are the control parameters for determining the weight of

trail intensity and length of path, respectively. N is the number of ants.

After selecting the next path, the trail intensity of pheromone is updated as:

$$\tau_{ij}(k+1) = \rho\tau_{ij}(k) + \Delta\tau_{ij} \quad (3)$$

In the above equation, ρ is a coefficient such that $(1-\rho)$ represents the evaporation of the trail between time k and k+1 and $\Delta\tau_{ij}$ is the amount of pheromone trail added to τ_{ij} by ants.

To apply the ACO algorithm for clustering, the following steps have to be taken:

Step 1: Generate the initial population and trail intensity

Step 2: Generate the initial population and trail intensity for ants in each colony (local search)

Step 3: Determine the next position

Step 4: Check the convergence condition

Simulated annealing algorithm: Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems. The concept is based on the manner in which liquids freeze or metals recrystallize in the process of annealing. In an annealing process, a melt is initially at high temperature and disordered and is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a frozen ground state at zero temperature. Hence, the process can be thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low, the system may become quenched (Tai-Hsi *et al.*, 2008).

The original Metropolis scheme was that an initial state of a thermodynamic system was chosen at energy E and temperature T. Holding T constant, the initial configuration is perturbed and the change in energy, ΔE , is computed. If the change in energy is negative, the new configuration is accepted. If the change in energy is positive, it is accepted with a probability factor $\exp(-\Delta E/T)$. This process is then repeated sufficient times to give good sampling statistics for the current temperature and then the temperature is decremented and the entire process repeated until a frozen state is achieved at $T = 0$.

The general procedure for the SA algorithm can be summarized as follows (Christober *et al.*, 2007):

Step 1: Select an initial solution X and an initial temperature T.

Step 2: Find another solution, namely X_{next} , by modifying the last answer X.

Step 3: Calculate the energy differential $\Delta E = f(X_{next}) - f(X)$

Step 4: If $\Delta E < 0$ go to Step 9.

Step 5: Generate a random number, namely R, between 0 and 1.

Step 6: If $R < \exp(-\Delta E/T)$ go to Step 9.

Step 7: Repeat Steps 2-6 for a number of optimization steps for the given temperature.

Step 8: If no new solution, X_{next} is accepted and go to Step 10.

Step 9: Decrease the temperature T, replace X with X_{next} and go to Step 2.

Step 10: Reheat the environment with setting T to a higher value.

Step 11: Repeat Steps 1 through 10 until no further improvement obtained.

APPLICATION OF ACO-SA TO CLUSTERING

This section presents a new hybrid approach based on combining the ant colony optimization and simulated annealing algorithms for the clustering problem. To apply the hybrid algorithm, named ACO-SA, on the clustering the following steps should be repeated:

Step 1: Generate an initial population: An initial population is randomly generated as follows:

$$\text{Population} = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_N \end{bmatrix} \quad (4)$$

$$X_i = [\text{Center}_1, \text{Center}_2, \dots, \text{Center}_k], \quad i = 1, 2, 3, \dots, N$$

$$\text{Center}_j = [x_1, x_2, \dots, x_d]$$

$$x_1^{\min} < x_1 < x_1^{\max}$$

where, Center_j is the j^{th} cluster center for the i^{th} individual. X_i is position of the i^{th} individual. d is the dimension of

each cluster center. X_i^{\max} and x_i^{\min} (each feature of center) are the maximum and minimum value of each point belonging to the j^{th} cluster center, respectively.

Step 2: Generate an initial trail intensity: At initialization phase, it is assumed that the trail intensity between each pair of ants is the same and is generated as follows:

$$\begin{aligned} \text{Trail_Intensity} &= [\tau_{ij}]_{N \times N} \\ \tau_{ij} &= \tau_0 \end{aligned} \tag{5}$$

where, τ_{ij} and τ_0 are trail intensity between the i^{th} and j^{th} ants and initial trail intensity, respectively.

Step 3: Determination of the next path: To determine the next path, at first, for each X_i a set A_i is defined by:

$$A_i = \{X \mid \|X_i - X_j\| \leq D(k), F(X_i) - F(X_j) \geq 0, X_i \neq X_j\} \tag{6}$$

where $D(k)$ is visibility. $\|\cdot\|$ is defined by:

$$\|X\| = \max |x_i|_{1 \leq i \leq n} \quad X = [x_1, x_2, \dots, x_n] \tag{7}$$

For determination of the next path, there are 2 cases:

Case 1: A_i is empty: In this case, the best solution (Y) is determined in the neighborhood of X_i by SA.

The neighborhood is defined as follows:

$$\{Y \mid \|X - Y\| \leq \alpha D(k)\} \tag{8}$$

where α is a positive parameter between 0 and 1.

The neighborhood of X_i is determined by simulated annealing as follows:

Step 1: Select an initial solution X and an initial temperature T .

Step 2: Find another solution, namely X_{next} , by modifying the last answer X .

Step 3: Calculate the energy differential $\Delta E = f(X_{\text{next}}) - f(X)$

Step 4: If $\Delta E < 0$ go to step 9.

Step 5: Generate a random number, namely R , between 0 and 1.

Step 6: If $R < \exp(-\Delta E/T)$ go to step 9.

Step 7: Repeat steps 2-6 for a number of optimization steps for the given temperature.

Step 8: If no new solution, X_{next} is accepted and go to step 10.

Step 9: Decrease the temperature T , replace X with X_{next} and go to step 2.

Step 10: Reheat the environment with setting T to a higher value.

Step 11: Repeat Steps 1 through 10 until no further improvement obtained.

After that SA finds the best local solution (Y), X_i is replaced by Y and then the trail intensity matrix is updated as follows:

$$\begin{aligned} \Delta\tau_{ij} &= r \quad \forall X_j \in \text{Population}, \\ X_i \neq X_j, \tau_{ij}(k+1) &= \rho\tau_{ij}(k) + \Delta\tau_{ij} \quad \forall i \neq j \end{aligned} \tag{9}$$

where r is a positive parameter between 0 and 1.

Case 2: A_i is not empty: In this case, selection of the next path is based on Eq. 2. Since in some optimization problems, L_{ij} is not known, we can define the inverse of L_{ij} as follows:

$$\phi_{ij} = F(X_i) - F(X_j) \quad \forall X_j \in A_i \tag{10}$$

In this algorithm, we consider X_i as a selection. In this case ϕ_{ii} and τ_{ii} are defined by:

$$\phi_{ii} = \frac{1}{m} \sum_{X_j \in A_i} \phi_{ij}, \quad \tau_{ii} = \frac{1}{m} \sum_{X_j \in A_i} \tau_{ij} \tag{11}$$

where m is the number of elements of A_i .

The transition probabilities are defined as:

$$P_0 = \frac{(\phi_{ii})^{\gamma_1} (\frac{1}{m} \sum_{X_j \in A_i} (\tau_{ij}))^{\gamma_2}}{(\phi_{ii})^{\gamma_1} (\frac{1}{m} \sum_{X_j \in A_i} (\tau_{ij}))^{\gamma_2} + \sum_{X_j \in A_i} (\phi_{ij})^{\gamma_1} (\tau_{ij})^{\gamma_2}} \tag{12}$$

$$P_{ij} = \frac{(\phi_{ij})^{\gamma_1} (\tau_{ij})^{\gamma_2}}{(\phi_{ii})^{\gamma_1} (\frac{1}{m} \sum_{X_j \in A_i} (\tau_{ij}))^{\gamma_2} + \sum_{X_j \in A_i} (\phi_{ij})^{\gamma_1} (\tau_{ij})^{\gamma_2}} \tag{13}$$

where γ_1 and γ_2 are the control parameters.

The roulette wheel is used for the stochastic selection. If the selection result is P_{ij} , then:

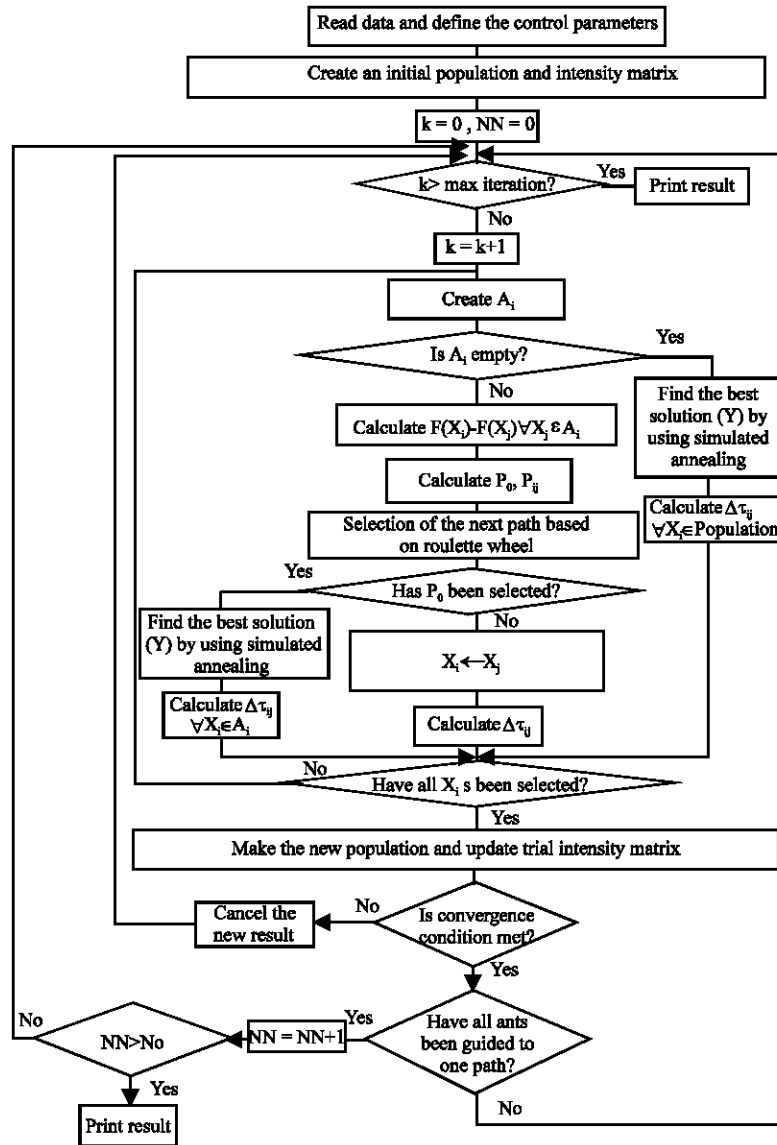


Fig. 2: Flowchart of the hybrid algorithm

$$\begin{aligned}
 X_i &= X_j \\
 \Delta\tau_{ij} &= P_{ij} \\
 \tau_{ij}(k+1) &= \rho\tau_{ij}(k) + \Delta\tau_{ij} \quad \forall X_j \in A_i
 \end{aligned}
 \tag{14}$$

$$\tau_{ij}(k+1) = \rho\tau_{ij}(k) + \Delta\tau_{ij} \quad \forall i \neq j, X_j \in A_i \tag{16}$$

If the selection result is P_0 , the best solution (Y) is found in the neighborhood of X_i by simulated annealing as mentioned in case 1 (A_i is empty). After that X_i is replaced by Y and then the trail intensity matrix is updated as follows:

$$\Delta\tau_{ij} = \frac{(F(X_i) - F(Y))^n \left(\frac{1}{m} \sum_{X_j \in A_i} (\tau_{ij})^{\gamma_2}\right)}{(\varphi_{ii})^n \left(\frac{1}{m} \sum_{X_j \in A_i} (\tau_{ij})^{\gamma_2}\right) + \sum_{X_j \in A_i} (\varphi_{ij})^n (\tau_{ij})^{\gamma_2}} \tag{15}$$

Step 4: Check of the convergence condition: After all ant colonies (X_i) find the next path, the convergence condition is checked by:

$$\begin{aligned}
 F(k+1) &\leq F(k) \\
 F(k+1) &= \frac{1}{N} \sum_{i=1}^N F(X_i) \quad \forall X_i \text{ New population} \\
 F(k) &= \frac{1}{N} \sum_{i=1}^N F(X_i) \quad \forall X_i \text{ Old population}
 \end{aligned}
 \tag{17}$$

If the convergence condition is not satisfied, cancel the new results and then go to step 3. If all ants are guided to

one path after NN iterations, stop and print the results. If the results are not changed after NNN iterations, increase the visibility ($D(k)$) and neighborhood of search (α). Here NN, NNN are coefficients. Figure 2 shows the flowchart of the hybrid algorithm.

RESULTS AND DISCUSSION

Experimental results comparing the ACO-SA clustering algorithm with several typical stochastic algorithms including the ACO, SA and K-means algorithms are provided for six real-life data sets (Iris, Wine, Vowel, Contraceptive Method Choice(CMC), Wisconsin breast cancer and Ripley's glass), which are described as follows: Iris data ($N = 150, d = 4, K = 3$): This is the iris data set. These data set with 150 random samples of flowers from the iris species *setosa*, *versicolor* and *virginica* collected by Anderson. From each species there are 50 observations for sepal length, sepal width, petal length and petal width in cm. This data set was used by Fisher in his initiation of the linear-discriminant-function technique.

Wine data ($N = 178, d = 13, K = 3$): This is the wine data set, which also taken from MCI laboratory. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value.

Contraceptive method choice ($N = 1473, d = 10, K = 3$): This data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Vowel data set ($N = 871, d = 3, K = 6$): This data set consists of 871 patterns. There are six overlapping vowel classes and three input features.

Wisconsin breast cancer ($N = 683, d = 9, K = 2$), which consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects).

Ripley's glass ($N = 214, d = 9, K = 6$), for which data were sampled from six different types of glass: building windows float processed (70 objects), building windows non-float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects) and headlamps (29 objects), each with nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron.

The algorithms are implemented by using Matlab 7.1 on a Pentium IV, 2.8 GHz, 512 GB RAM computer. The parameters required for implementation of the ACO-SA algorithm are $\gamma_1, \gamma_2, \rho, D_0$ and T . In this study, the best values for the aforementioned parameters are $\gamma_1 = \gamma_2 = 1.0, \rho = 0.99, D_0 = 10$ and $T = 100$ determined by 10 runs of the algorithm.

Table 1-6 present a comparison among the results of ACO-SA, ACO (Fathian *et al.*, 2007; Shelokar *et al.*, 2004), SA (Fathian *et al.*, 2007) and K-means (Kao *et al.*, 2008) for 100 random tails on the mentioned data sets.

For Iris data (Table 1) it is found that the ACO-SA clustering algorithm provides the optimal value of 96.6602, most of the times. The ACO, SA and K-means algorithms attain 96.753, 97.4573 and 97.333 respectively. The standard deviation of the fitness function for ACO-SA clustering algorithm is 0.12196, which it significantly is smaller than other methods. The number of function evaluations of ACO-SA, ACO, SA and K-means are 3,629, 4,931, 5314 and 120, respectively. The best value obtained by ACO-SA clustering algorithm for Wine data (Table 2) is 16,298.628 (which is obtained in 90% of the total runs). The ACO, SA and K-means algorithms, on the other hand, fail to attain this value in any of their runs. From Table 3 for CMC data, it is found that the ACO-SA was able to find the optimum in most of runs. The ACO-SA clustering algorithm attains 5,696.6075 that it is better than the best value of ACO, SA and K-means algorithms (5,701.923, 5,849.038 and 5,842.20). For the vowel data set (Table 4), the best global solution, the worst global solution, the average and the standard deviation of the ACO-SA are 148,978.6294, 149,183.1834, 149,057.8251 and 75.539868, respectively. In terms of the number of function evaluations, K-means needs the least number of function evaluations. The results for Wisconsin breast cancer data are shown in Table 5. The best value that the ACO-SA attains is 2,965.97 and the worst is 2,969.01, both of which are better than those obtained by ACO, SA and K-means algorithms. For Ripley's glass Data, the ACO-SA algorithm attains the best value of 199.89. The results of other algorithms are inferior to that of ACO-SA. The number of function evaluations of ACO-SA is less than ACO and SA but it is greater than K-means algorithm.

Table 1: Results obtained by the five algorithms for 100 different runs on Iris data

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	96.6602	96.73192	96.86381	0.12196	3629
ACO	96.753	97.453	98.023	0.567	4,931
SA	97.4573	99.957	102.01	2.018	5314
K-means	97.333	106.05	120.45	14.6311	120

Table 2: Results obtained by the five algorithms for 100 different runs on Wine data

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	16,298.628	16,310.283	16,322.438	10.62197	11,628
ACO	16,346.7820	16,417.127	16,502.943	80.3731	15,473
SA	16,473.4825	17,521.094	18,083.251	753.084	17,264
K-means	16,555.68	18.061	18,563.12	793.213	390

Table 3: Results obtained by the five algorithms for 100 different runs on CMC data

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	5,696.6075	5,698.2618	5,700.2681	1.98238	11,628
ACO	5,701.923	5,819.1347	5,912.4300	45.6347	20,436
SA	5,849.038	5,893.4823	5,966.947	501.8672	26,829
K-means	5,842.20	5,893.60	5,934.43	473.16	270

Table 4: Results obtained by the five algorithms for 100 different runs on Vowel data

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	148,978.6294	149,057.8251	149,183.1834	75.539868	6,829
ACO	149,201.632	161,431.0431	165,804.671	2746.0416	8,436
SA	149,468.268	162,108.5381	165,996.428	2846.23516	9,528
K-means	149,422.26	15,9242.89	161,236.81	916	180

Table 5: Results obtained by the five algorithms for 100 different runs on Wisconsin breast cancer

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	2,965.97	2,967.04	2,969.01	1.716804	10,098
ACO	2,970.49	3,046.06	3,242.01	90.50028	15,983
SA	2,992.53	3,086.06	3,451.38	111.0102	17,853
K-means	2,987.19	2,987.78	2,988.24	0.38	180

Table 6: Results obtained by the five algorithms for 100 different runs on Ripley's glass

Method	Function value			Standard deviation	No. of function evaluations
	F_{best}	$F_{average}$	F_{worst}		
ACO-SA	199.89	202.56	205.29	2.013824	55,694
ACO	269.72	273.46	280.08	3.584829	196,581
SA	273.27	276.21	285.93	4.976188	199,468
K-means	215.74	12.471	255.38	235.50	630

The simulation results in the Table 6 demonstrate that the proposed hybrid evolutionary algorithm converges to global optimum with a smaller standard deviation and less function evaluations and leads naturally to the conclusion that the ACO-SA algorithm is a viable and robust technique for data clustering.

CONCLUSION

In this study a new hybrid evolutionary algorithm is proposed to find optimal or near optimal solutions for clustering problems of allocating N objects to k clusters.

The proposed algorithm is based on a combination of ACO and SA algorithms. The ACO-SA algorithm is tested on several well known real life datasets. The experimental results indicate that the proposed optimization algorithm is at least comparable to the other algorithms in terms of function evaluations and standard deviations. Regardless of robustness and efficiency of ACO-SA algorithm, it is applicable when the number of clusters is known a priori. The results illustrate that the proposed ACO-SA optimization algorithm can be considered as a viable and an efficient heuristic.

REFERENCES

- Amir, A. and D. Lipika, 2007. A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowledge Eng.*, 63: 503-527.
- Christober, C., A. Rajan and M.R. Mohan, 2007. An evolutionary programming based simulated annealing method for solving the unit commitment problem. *Int. J. Elect. Power Energy Syst.*, 29: 540-550.
- Dorigo, M., M. Birattari and T. Stutzle, 2006. Ant colony optimization. *IEEE Comput. Intel. Mag.*, 1: 28-39.
- Fathian, M., B. Amiri and A. Maroosi, 2007. Application of honey-bee mating optimization algorithm on clustering. *Applied Math. Comput.*, 190: 1502-1513.
- Ho, S.L., Y. Shiyou, N. Guangzheng and J.M. Machado, 2006. A modified ant colony optimization algorithm modeled on tabu-search methods. *IEEE Trans. Magnet.*, 4: 1195-1198.
- Huang, S.J., 2001. Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches. *IEEE Trans. Energy Convers.*, 16: 296-301.
- Kao, Y.T., Z. Erwie and I.W. Kao, 2008. A hybridized approach to data clustering. *Expert Syst. Appl.*, 34: 1754-1762.
- Kwang, M.S. and W.H. Sun, 2003. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Trans. Syst. Man Cybernetics, Part A*, 33: 560-572.
- Lu, J.F., J.B. Tang, Z.M. Tang and J.Y. Yang, 2008. Hierarchical initialization approaches for K-Means clustering. *Pattern Recogn. Lett.*, 29: 787-795.
- Merkle, D., M. Middendorf and H. Schmech, 2002. Ant colony optimization for resource constrained project scheduling. *IEEE Trans. Evolut. Comput.*, 6: 333-346.
- Niknam, T., A.M. Ranjbar and A.R. Shirani, 2005a. A new approach for distribution state estimation based on ant colony algorithm with regard to distributed generation. *J. Intell. Fuzzy Syst.*, 16: 119-131.
- Niknam, T., A.M. Ranjbar and A.R. Shirani, 2005b. A new approach based on ant algorithm for Volt/Var control in distribution network considering distributed generation. *Iran. J. Sci. Technol. Trans. B*, 29: 1-15.
- Shelokar, P.S., V.K. Jayaraman and B.D. Kulkarni, 2004. An ant colony approach for clustering. *Anal. Chim. Acta*, 509: 187-195.
- Tai-Hsi, W., C. Chin-Chih and C. Shu-Hsing, 2008. A simulated annealing algorithm for manufacturing cell formation problems. *Expert Syst. Appl.*, 34: 1609-1619.