World Applied Sciences Journal 4 (2): 300-307, 2008
ISSN 1818-4952

An Efficient Hybrid Evolutionary Algorithm for Cluster Analysis

[1]*Taher Niknam,* [2]*Bahman Bahmani Firouzi and* [1]*Majid Nayeripour*

[1]Department of Electrical Engineering, Shiraz University of Technology, Shiraz, Iran
[2]Marvdasht Islamic Azad University, Marvdasht, Fars, Iran

Abstract: Clustering problems appear in a wide range of unsupervised classification applications such as pattern recognition, vector quantization, data mining and knowledge discovery. The k-means algorithm is one of the most widely used clustering techniques. Unfortunately, k-means is extremely sensitive to the initial choice of centers and a poor choice of centers may lead to a local optimum that is quite inferior to the global optimum. This paper presents an efficient hybrid evolutionary optimization algorithm based on combining Ant Colony Optimization (ACO) and Simulated Annealing (SA), called ACO-SA, for cluster analysis. The performance is evaluated through several benchmark data sets. The simulation results show that the proposed algorithm outperforms the previous approaches such as SA, ACO and k-means for partitional clustering problem.

Key words: Ant Colony Optimization (ACO) . K-means clustering . Data clustering . Hybrid evolutionary optimization algorithm . Simulated Annealing (SA)

INTRODUCTION

Clustering has long been the basis of many knowledge discovery tasks such as machine learning, statistics, data mining and pattern recognition. There are two main branches of clustering; Hierarchical and Partitional [1-6]. Partitional clustering involves partitioning a given set of data points into a number of distinct groups, termed clusters [2, 3]. In hierarchical clustering, the output is a tree showing a sequence of clustering with each cluster being a partition of the data set [1, 4]. In this paper we concentrate on the partitional clustering. The k-means clustering is one of the popular partitional clustering methods. The k-means algorithm partitions a set of objects into k subsets and is one of the most popular and widely used clustering techniques because it is easy to implement and very efficient. However, the k-means algorithm suffers from several drawbacks. The objective function of the k-means is not convex and hence it may contain many local minima. Consequently, in the process of minimizing the objective function, there exists a possibility of getting stuck at local minima, as well as at local maxima and saddle points. The outcome of the k-means algorithm, therefore, heavily depends on the initial choice of the cluster centers. In order to overcome local optima problem lots of studies have done in clustering. So, in recent years, many clustering algorithms based on evolutionary algorithms such as genetic algorithm, tabu search and SA have been introduced. For instance, a genetic algorithm for selecting centers to seed the popular k-means method for clustering has been used by Laszlo *et al.* [7]. A hybrid technique based on combining the k-means algorithm, Nelder-Mead simplex search and particle swarm optimization has been proposed for the clustering problem by Kao *et al*. [8]. Application of honeybee mating optimization in clustering has been presented by Fathian *et al*. [9]. Application of the ant colony optimization on clustering analysis in data mining has been used by Kuo *et al*. [10] and Sheokar *et al*. [11]. A tabu search algorithm for clustering problem has been proposed by Michael *et al*. [12], Sung *et al*. [13] and Khaled [14].

The ant colony optimization (ACO) paradigm was first proposed by Dorigo [15]. Since then, a number of studies based on ant colony algorithm have been carried out in order to solve some optimization problems such as traveling salesman problem, unit commitment, etc [15, 20]; however, in these studies, the search domains are in discrete form. Also a number of methods based on ant colony have been presented to solve the problem in both continuous and discrete search domains such as economic dispatch, reactive power pricing in restructured networks [19, 24]. For instance, Wang *et al*. have presented a hybrid evolutionary algorithm based on combining the ACO and SA algorithms. They have used the output of the ACO algorithm as the initial point of the SA algorithm. In this paper, a new hybrid

Corresponding Author: Dr. Taher Niknam, Department of Electrical and Electronic, Shiraz University of Technology, Modares Blvd, P.O. 71555-313, Shiraz, Iran

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 for each colony in the ACO algorithm. The proposed hybrid algorithm is applied for optimally clustering *N* object into *K* clusters. The simulation results illustrate that this algorithm not only has a better response but also converges more quickly than the ordinary evolutionary methods like ACO and SA.

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:

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

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, $1 = i = n$, $1 = j = K$ 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 [8]:

- Place *K* points into the space represented by the objects that are being clustered. These points represent the initial group centroids.
- Assign each object to the group that has the closest centroid.
- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat steps 2 and 3 until a certain criterion is met, such as the centroids no longer moving or a preset 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:

$$P_{ij} = \frac{(t_{ij})^{\gamma_2}(1/L_{ij})^{\gamma_1}}{\sum_{\substack{j=1 \\ j \neq i}}^{N} (t_{ij})^{\gamma_2}(1/L_{ij})^{\gamma_1}} \quad (2)$$

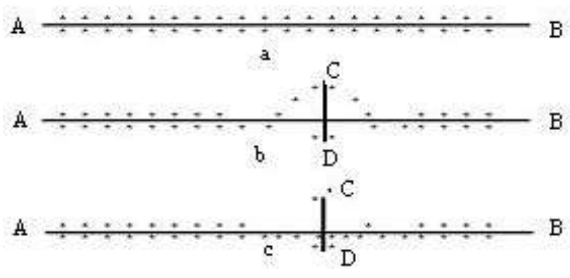

Fig. 1: An example of finding the shortest path by ants

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 [23]:

Step1: 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 recrystalize 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 [25, 26].

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 $T=0$.

APLLICATION OF ACO-SA TO CLUSTERING

In this section, a new hybrid approach based on the ant colony algorithm and the simulated annealing algorithm is presented to solve the clustering problem. In order to follow this goal, a number of N colonies are considered. To choose a movement direction, each colony needs to find the best local and global positions as follows:

Finding the best global position: Suppose the i^{th} colony wants to change its position. At first, the transition probabilities between the i^{th} and j^{th} colonies are calculated as indicated in (4).

$$[P_{Gn}]_i = [P_{Gi1}, P_{Gi2}, \ldots, P_{GiN}]_{1*N}$$
$$P_{Gij} = \frac{(\tau_{Gij})^{\gamma_2}(1/L_{ij})^{\gamma_1}}{\sum_{j=1}^{N}(\tau_{Gij})^{\gamma_2}(1/L_{ij})^{\gamma_1}} \quad (4)$$

where P_{Gij} is the transition probability between the i^{th} and j^{th} individuals.
The cumulative probabilities are calculated as:

$$[C_{Gn}]_i = [C_{G1}, C_{G2}, \ldots, C_{GN}]_{1*N}$$
where
$$C_{G1} = P_{Gi1}$$
$$C_{G2} = C_{G1} + P_{Gi2}$$
$$\ldots.$$
$$C_{Gj} = C_{Gj-1} + P_{Gij} \quad (5)$$
$$\ldots.$$
$$C_{GN} = C_{GN-1} + P_{GiN}$$

In above equations, C_{Gj} is cumulative probability for the j^{th} individual. The roulette wheel is used for stochastic selection of the best global position as follows:

A number between 0 and 1 is randomly generated and compared with the calculated cumulative probabilities. The first term of cumulative probabilities (C_{Gj}), which is greater than the generated number, is selected and the associated position is considered as the best global position.

Finding the best local position: By using the simulated annealing algorithm, the best local position is found around the i^{th} colony.

Determination of the next position: The movement direction for each colony is obtained from linear

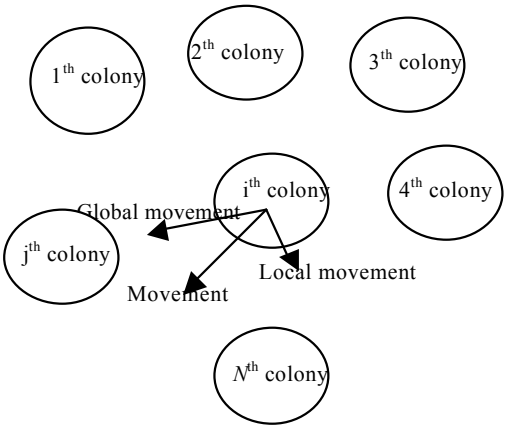

Fig. 2: Determination of the movement direction for colonies

combination of the best global and local positions. Fig. 2 shows the above-mentioned procedure graphically.

To apply the hybrid algorithm named ACO-SA on clustering the following steps should be repeated:

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

$$Population = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_N \end{bmatrix}$$

$$X_i = [Center_1, Center_2 \dots Center_K], \quad i = 1,2,3,\dots,N$$
$$Center_j = [x_1, x_2, \dots, x_d] \quad (6)$$
$$x_i^{min} < x_i < x_i^{max}$$

where $Center_j$ is the j^{th} cluster center for the i^{th} individual. X_i is the 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 initial trail intensity: At initialization phase, it is assumed that the trail intensity between each pair of colonies is the same and is generated as follows:

$$Trail\_Intensity = [\tau_{ij}]$$
$$\tau_{ij} = \tau_0 \quad (7)$$

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

Determination of the next path: Assume that the i^{th} colony wants to determine its next position. As mentioned before, the movement direction of each colony is a linear combination of the best global and local positions, which can be selected as:

Selection of the best global position: The best global position (X_{Global}) is found based on equations (4) and (5). Since L_{ij}, called desirability factor, is not known in the clustering problem, we can define its inverse as follows:

$$1/L_{ij} = \phi_{Gij} = f(X_i) - f(X_j); \quad j \neq i \quad (8)$$

$f(X_i)$ and $f(X_j)$ are the objective function values of the clustering problem for the i^{th} and j^{th} colonies.

The transition probabilities between the i^{th} colony and the rest of the colonies are defined as

$$P_{Gij} = \frac{(\phi_{Gij})^{\gamma_1} (\tau_{Gij})^{\gamma_2}}{\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} (\phi_{Gij})^{\gamma_1} (\tau_{Gij})^{\gamma_2}\right)}, \quad j = 1,2,\dots, N; \quad i \neq j \quad (9)$$

The cumulative probabilities for colonies are calculated based on the calculated transition probabilities. The best global position is selected by the roulette wheel. The global trail intensities are updated as follows:

$$\Delta\tau_{Gij} = P_{Gij}$$
$$\tau_{Gij}(k+1) = \rho\tau_{Gij}(k) + \Delta\tau_{Gij} \quad (10)$$

Applying the SA algorithm to find the best local position: The simulated annealing is used to find the best local position (X_{Local}) around the i^{th} colony as follows:

1: Select an initial solution X and an initial temperature T.
2: Find another solution, namely X_{next}, by modifying the last answer X.
3: Calculate the energy differential $\Delta E = f(X_{next}) - f(X)$
4: If $\Delta E < 0$ go to Step 9.
5: Generate a random number, namely R, between 0 and 1.
6: If $R < \exp(-\frac{\Delta E}{T})$ go to Step 9.
7: Repeat Steps 2–6 for a number of optimization steps for the given temperature.

8: If no new solution, X_{next} is accepted and go to Step 10.
9: Decrease the temperature *T*, replace *X* with X_{next} and go to Step 2.
10: Reheat the environment with setting T to a higher value.
11: Repeat Steps 1 through 10 until no further improvement obtained.

Determination of the next position: After selecting the best local and global positions, the next position is determined as follows:

$$X_i(k+1) = X_i(k) + \text{rand}()*(X_{Local} - X_i(k)) + \text{rand}()*(X_{Global} - X_i(k)) \quad (11)$$

It must be noted that in the new position the constraint must be completely satisfied.

Step 4: Check of convergence: After that all colonies have found their next positions, the convergence condition is checked as below:

$$\sqrt{\sum_{i=1}^{N} \left| X_i^{k+1} - X_i^{k} \right|^2} < \varepsilon \quad (12)$$

where $k+1$ is the current iteration.

If the convergence condition is satisfied, the task is complete and if not, the process must be repeated from step 3. Figure 3 shows the complete flowchart of the process.

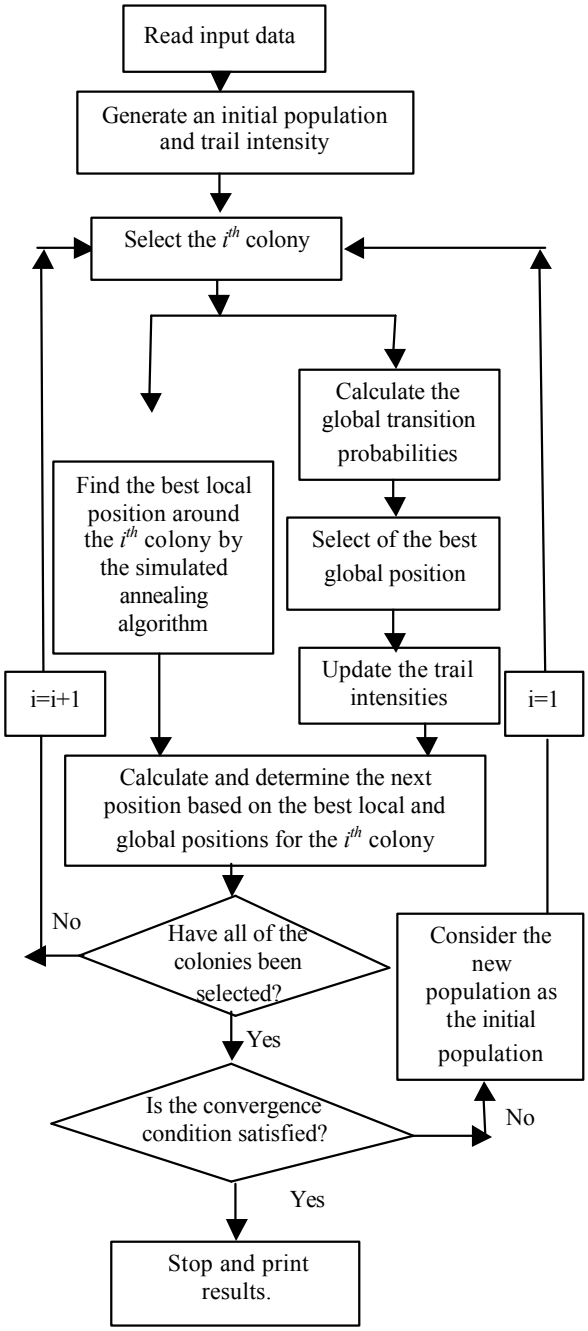

Fig. 3: The flowchart of the ACO-SA algorithm

EXPERIMENTAL RESULTS

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 (1935). 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 (1936) 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

Table 1: Result obtained by the four algorithms for 100 different runs on Iris data

	ACO-SA	ACO	SA	K-means
Best solution	96.66020	96.753	97.4573	97.333
Average value	96.73192	97.453	99.9570	106.050
Worst solution	96.86381	98.023	102.0100	120.450
Standard deviation	0.12196	0.567	2.0180	14.6311
# function evaluation	3629.0000	4,931.000	5314.0000	120.000

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

	ACO-SA	ACO	SA	K-means
Best solution	16,298.630	16,346.78	16,473.48	16,555.68
Average value	16,310.280	16,417.13	17,521.09	18,061.00
Worst solution	16,322.440	16,502.94	18,083.25	18,563.12
Standard deviation	10.62197	80.3731	753.084	793.213
# function evaluation	11,628.000	15,473.00	17,264.000	390.000

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

	ACO-SA	ACO	SA	K-means
Best solution	5,696.61	5,701.92	5,849.04	5,842.20
Average value	5,698.26	5,819.13	5,893.48	5,893.60
Worst solution	5,700.27	5,912.43	5,966.95	5,934.43
Standard deviation	1.98238	45.6347	501.8672	473.16
# function evaluation	11,628.00	20,436.00	26,829.00	270.00

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. Table 1-6 present a comparison among the results of ACO-SA, ACO, SA and K-means for 100 random runs on the mentioned data sets.

The simulation results given in Table 1-6 show that ACO-SA is very precise as provides the optimum value and small standard deviation in comparison with other methods. The results obtained on the iris data set indicate that ACO-SA converges to the global optimum of 96.6602 in most of the times while the best solutions of ACO, SA and K-means are 96.7531, 97.4573 and 97.333, respectively. The standard deviation of the fitness function for this algorithm is 0.12196. For wine data set (Table 2) the optimum value is 16,298.628, which is obtained in 90% the runs of ACO-SA algorithm. Noticeably, other algorithms fail to attain this value even once within 100 runs. Table 3 provides the results of the algorithms on the CMC data set. Results of the ACO-SA algorithm are far superior to those of others. For the vowel data set, 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, 149,057.8251and 75.539868, respectively. For Wisconsin breast cancer data set (Table 5) the ACO-SA provide the optimum value of 2,965.97 while the ACO, SA and k-means algorithms attain 2,970.49, 2,992.53 and 2,987.19, respectively. The ACO-SA was able to find the optimum in most of the times. Finally, Table 6 shows the best, average, worst and standard deviation values

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

	ACO-SA	ACO	SA	K-means
Best solut ion	148,978.63	149,201.63	149,468.27	149,422.26
Average value	149,057.83	161,431.04	162,108.54	159,242.89
Worst solution	149,183.18	165,804.67	165,996.43	161,236.81
Standard deviation	75.539868	2746.0416	2846.23516	916.00
# function evaluation	6,829.00	8,436.00	9,528.00	180.00

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

	ACO-SA	ACO	SA	K-means
Best solution	2,965.97	2,970.49	2,992.53	2,987.19
Average value	2,967.04	3,046.06	3,086.06	2,987.78
Worst solution	2,969.01	3,242.01	3,451.38	2,988.24
Standard deviation	1.716804	90.50028	111.0102	0.38
# function evaluation	10,098.0000	15,983.000	17,853.0000	180.00

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

	ACO-SA	ACO	SA	K-means
Best solution	199.89	269.72	273.27	215.74
Average value	202.56	273.46	276.21	12.47
Worst solution	205.29	280.08	285.93	255.38
Standard deviation	2.013824	3.584829	4.976188	235.50
# function evaluation	55,694.00	196,581.0000	199,468.0000	630.00

attained by algorithms for Ripley's glass data set. The ACO-SA clustering algorithm is able to provide the same partition of the data points in all runs. As earlier, the results of the other algorithms are inferior to that of ACO-SA. K-means in terms of the number of function evaluations needs the least number of function evaluations, but the results are less than satisfactory. For the iris data set, the number of function evaluations of ACO-SA, ACO, SA and K-means are 3,629, 4,931, 5314 and 120, respectively. The number of function evaluations of ACO-SA for Wine, CMC, Vowel, Wisconsin breast cancer and Ripley's glass are 11,628, 11,628, 6,829, 10,098 and 55,694, respectively. These results indicate that the number of function evaluations of ACO-SA is less than those of other evolutionary algorithms. The simulation results in the tables demonstrate that the proposed hybrid evolutionary algorithm converges to a global optimum with a smaller standard deviation and fewer function evaluations and leads naturally to the conclusion that the ACO-SA algorithm is a viable and robust technique for data clustering.

CONCLUSION

A hybrid evolutionary optimization algorithm to solve clustering problems has been developed in this paper. The algorithm is based on combination of the ant colony optimization and the simulated annealing. In the proposed algorithm, we have used the simulated annealing algorithm as a local searcher for each colony. To evaluate the performance of the hybrid algorithm, it is compared with other stochastic algorithms viz. the original ACO, SA and k-means algorithms on several well known real life data sets. 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 result illustrate that the proposed ACO-SA optimization algorithm can be considered as a viable and an efficient heuristic.

REFERENCES

1. Lu, J.F., J.B. Tang, Z.M. Tang and J.Y. Yang, 2008, Hierarchical initialization approach for k-means clustering, Pattern Recognition Letters, 29 (6): 787-795.
2. Zhou Hong and Y. Liu, 2008. Accurate integration of multi-view range images using k-means clustering, Pattern Recognition, 41 (1): 152-175.

3. Ahmad, A. and L. Dey, 2007, A k-mean clustering algorithm for mixed numeric and categorical data, Data & Knowledge Engineering, 63 (2): 503-527.
4. Chung, H.C., C.C. Long and S.Y. Wei, 2007. Hierarchical clustering of mixed data based on distance hierarchy, Information Sciences, 177 (20): 4474-4492.
5. Redmond, J. Stephen and C. Henghan, 2007. A method for initializing the K-means clustering algorithm using kd-trees. Pattern Recognition Letters, 28 (8): 965-973.
6. Papmichail, P.G. and P.D. Papmichail, 2007. The k-means range algorithm for personalized data clustering in e-commerce. European Journal of Operational Research, 177 (3): 1400-1408.
7. Laszlo, M. and M. Sumitra, 2007. A genetic algorithm that exchanges neighboring centers for k-means clustering. Pattern Recognition Letters, 28 (16): 2359-2366.
8. Kao, Y.T., E. Zahara and I.W. Kao, 2008. A hybridized approach to data clustering, Expert Systems with Applications, 34 (3): 1754-1762.
9. Mohammad, F., B. Amiri and A. Marsooi, 2007. Application of honey-bee mating optimization algorithm on clustering, Applied Mathematics and Computation, 190 (2): 1502-1513.
10. Kuo, R.J., H.S. Wang, T. Tung and S.H. Chou, 2005. Application of ant k-means on clustering analysis in data mining. Computers & Mathematics with Applications, 50 (10): 1709-1724.
11. Shelokar, P.S., V.K. Jayraman and Kulkami, 2004. An ant colony approach for clustering, Analytica Chimica Acta, 509 (2): 187-195.
12. Michael, K.N. and J.C. Wong, 2002. Clustering categorical data sets using tabu search techniques, Pattern Recognition, 35 (12): 2783-2790.
13. Sung, C.S. and H.W. Jin, 2000. A tabu-search-based heuristic for clustering, Pattern Recognition, 33 (5): 849-858.
14. Khaled, S.A., 1995. A tabu search approach to the clustering problem. Pattern Recognition, 28 (9): 1443-1451.
15. Dorigo, M., G.D. Caro and L.M. Gambardella, 1999. Ants algorithms for discrete optimization, Artificial Life, 5 (3): 137-172.
16. Krishnoiger, K. and S.H. Cheraghi, 2002. Ant algorithms: Review and future application, International Engineering Research Conference.
17. Huang, S.J., 2001. Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches, IEEE Trans. On Energy Conversion, 16 (3): 296-301.
18. Merkle, D., M. Middendorf M. and H. Schmech, 2002. Ant colony optimization for resource constrained project scheduling, IEEE Trans. On Evolutionary Computation, 6 (4): 333-346.
19. Bauer, A., B. Bullnheimer, R.F. Hartl and C. Strauss, 1999. A ant colony optimization approach for single machine total tardiness problem. Proceedings of the 1999 Congress on Evolutionary Computation, CEC, 99 (2): 1445-1450.
20. Sisworahardjo, N.S. and A.A.E. Keib, 2002. Unit commitment using the ant colony search algorithm, 2002 Large Engineering Systems Conference on Power Engineering, LESCOPE 02; pp: 2-6.
21. Hou, Y.H., Y.W. Wu, L.J. Li and X.Y. Xiong., 2002. Generalized ant colony optimization for economic dispatch of power systems, Power Con 2002. Proceedings of International Conference on Power Systems Technology, 1: 225-229.
22. Niknam, T., H. Arabian and M. Mirjafari, 2004. Reactive power pricing in deregulated environments using novel search methods. Proceeding of the 2004 International Conference on Machine Learning and Cybernetics, 7: 4234-4239.
23. Niknam, T., A.M. Ranjbar and A.R. Shirani, 2005. A new approach for distribution state estimation based on ant colony algorithm with regard to distributed generation, Journal of Intelligent and Fuzzy Systems, 16 (2): 119-131.
24. Wang, G., W. Gong and R. Kastner, 2006. Application partitioning on programmable platforms using the ant colony optimization. Journal of Embedded Computing, 2 (1): 1-19.
25. Wu, T.H., C.C. Chang and S.H. Chung, 2008. A simulated annealing algorithm for manufacturing cell formation problems. Expert Systems with Applications, 34 (3): 1609-1617.
26. Meer, K., 2007. Simulated annealing versus metropolis for a TSP instance, Information Processing Letters, 104 (6): 216-219.