Scientific Research and Essays Vol. 6(34), pp. 6821-6828, 30 December, 2011
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.523
ISSN 1992-2248 ©2011 Academic Journals

Full Length Research Paper

On using tabu search for fuzzy clustering analysis

Yongguo Liu[1,2,3,4]*, Xindong Wu[3] and Yidong Shen[2]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China,
Chengdu 611731, P. R. China.
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences,
Beijing 100191, P. R. China.
[3]Department of Computer Science, University of Vermont, Burlington, Vermont 05405, USA.
[4]Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan University,
Xiangtan 411105, P. R. China.

Accepted 29 August, 2011

Clustering is an important technique for discovering the inherent structure in a given data set without any 'priori' knowledge. Fuzzy clustering analysis is to assign objects to a given number of clusters with respect to some criteria such that each object may belong to more than one cluster with different degrees of membership. In this article, a new fuzzy clustering method based on tabu search called Improved Tabu Search Fuzzy Clustering (ITSFC) is proposed to find the proper clustering of data sets. In the ITSFC approach, a fuzzy *c*-means operation is developed to fine-tune the clustering solution obtained in the process of iterations and a divide-and-merge operation is designed to establish the neighborhood. Experimental results on two artificial and four real life data sets are given to illustrate the superiority of the proposed algorithm over a tabu search clustering algorithm and an artificial bee colony clustering algorithm.

Key words: Fuzzy clustering, tabu search, artificial bee colony.

INTRODUCTION

Clustering is an unsupervised process that divides a given set of objects into groups so that objects within a cluster are similar with one another and dissimilar with the objects in other clusters. It has been applied across many disciplines such as machine learning, pattern recognition and statistics (Pedrycz, 2005; Xu and Wunsch, 2008). To date, many clustering algorithms have been reported and they can be divided into two main categories: hierarchical and partitional (García-Escudero et al., 2010; Omran et al., 2007).

In this article, we focus our attention on partitional clustering. Partitional clustering algorithms determine the clustering solution by maximizing the similarities among objects within the same group while minimizing the dissimilarities between different groups. Among partitional clustering approaches, *k*-means a typical iterative hill-climbing hard clustering method is popular (Liu et al., 2008; Selim and Ismail, 1984). It is known that hard clustering algorithms assign each object to one and only one cluster which are inappropriate for the data sets where the boundaries between clusters may not be well defined (Amiri et al., 2009; Chang et al., 2009; Jarhoui et al., 2007; Laszlo and Mukherjee, 2006; Zhang et al., 2010). In such cases, fuzzy clustering will be a better choice for grouping data sets.

In fuzzy clustering analysis, each object may belong to more than one cluster and its membership grade represents the degree to which the object belongs to a particular cluster. It is known that the Fuzzy *C*-means algorithm (FCM) (Baraldi and Blonda, 1999; Bezdek, 1981; García-Escudero et al., 2010) is one of the most frequently used fuzzy clustering methods. However, it is sensitive to initial clusters and can be trapped into local optima. Improper initialization would lead the FCM algorithm to produce inappropriate output of clusters. As the partitional clustering task can be stated as an

*Corresponding author. E-mail: liuyg_cn@163.com.

optimization problem, recently metaheuristic techniques are employed to deal with the fuzzy clustering problem so as to achieve the optimal or near-optimal solution within a specified number of iterations (Izakian and Abraham, 2011; Kanade and Hall, 2007; Supratid and Kim, 2009). In this study, our aim is to develop an improved tabu search fuzzy clustering algorithm, compare it with another tabu search fuzzy clustering method and an artificial bee colony fuzzy clustering method, and to demonstrate the usefulness and effectiveness of the proposed approach. In the proposed algorithm, on one hand, a fuzzy *c*-means operation is employed to incorporate the domain knowledge in the clustering procedure so as to enhance the convergence speed of the clustering algorithm, and on the other hand, a divide-and-merge operation is designed to modulate the object distribution among different clusters so as to establish the set of neighboring solutions. As a result, a new tabu search clustering method is given called improved tabu search fuzzy clustering (ITSFC). Experimental results on two artificial and four real life data sets are reported to illustrate that the ITSFC algorithm can provide better objective function values than the other two fuzzy clustering approaches.

The rest of this article is organized as follows: subsequently, the fuzzy clustering problem and the related work are reported; then the ITSFC algorithm and its components are described in detail. Performance comparison between the ITSFC algorithm and some known clustering methods is then conducted on two artificial and four real life data sets. Finally, experimental results are analyzed and concluded.

Related work

In this article, we consider the fuzzy clustering problem defined as follows:

$$\begin{cases} \min J = \sum_{k=1}^{N}\sum_{i=1}^{C} \mu_{ik}^{m} \| \mathbf{x}_k - \mathbf{c}_i \|^2 \\ 0 \le \mu_{ik} \le 1 \qquad i = 1,\dots,C \text{ and } k = 1,\dots,N \\ \sum_{i=1}^{C} \mu_{ik} = 1 \qquad k = 1,\dots,N \\ 0 < \sum_{k=1}^{N} \mu_{ik} < N \qquad i = 1,\dots,C \end{cases}, \qquad (1)$$

where $\| \|^2$ denotes the squared Euclidean distance between object $\mathbf{x}_k$ and cluster center $\mathbf{c}_i$, N denotes the number of objects, C denotes the number of clusters, μ_{ik} denotes the membership degree of object k with respect to cluster i, and m denotes the fuzzy index that governs the influence of membership grades and is set to 2 here.

As the FCM algorithm tends to converge to local optima, researchers employed some metaheuristic techniques to solve the clustering problem under consideration. Chen et al. (2010) combined the genetic algorithm with fuzzy *c*-means to solve the fuzzy clustering problem. Their method employs optimal colony selection, stochastic match crossover and parallel structure mutation to evolve the proper clustering result. Kanade and Hall (2007) developed an ant colony optimization inspired approach to group data which is composed of two stages. Ants first move the cluster centers in the feature space, and then the best cluster centers found are used as the initial cluster centers for the FCM algorithm. Supratid and Kim (2009) reported a modified fuzzy ant clustering method by combining fuzzy *c*-means with the genetic algorithm and the ant colony system. The proposed method was employed in creating fuzzy color histograms in image retrieval. Mehdizadeh et al. (2008) presented a fuzzy clustering method called fuzzy particle swarm optimization (FPSO) which is based on particle swarm optimization and fuzzy *c*-means to solve the fuzzy clustering problem. Experimental results showed that the FPSO method is superior to the FCM method in terms of the objective function value. Karaboga and Ozturk (2010) recently introduced an artificial bee colony algorithm to fuzzy clustering analysis and proposed a new fuzzy clustering approach called artificial bee colony fuzzy clustering (ABC-FC). By designing three groups of bees, employed bees, onlooker bees and scout bees to improve the clustering solution; the authors showed the superiority of the ABC-FC algorithm over the FCM algorithm. Like optimizing other numerical test functions (Karaboga and Akaya, 2009), the authors just applied the artificial bee colony algorithm to minimize the objective function of the fuzzy clustering problem. No more attempts were made to deal with the problems in fuzzy clustering such as object distribution optimization.

Tabu search is a metaheuristic method that guides the local heuristic search procedure to explore the solution space beyond the local optimality (Gendreau and Potvin, 2010; Glover and Laguna, 1997). Al-sultan and Fedjki (1997) introduced tabu search to solve the fuzzy clustering problem called tabu search fuzzy clustering (TSFC) in this paper. The authors adopted three directions to discretize the solution moves and create the trial centers so as to create neighboring solutions. In addition, the probability threshold is used to moderate the shake-up on the current solution. Experimental results showed that the TSFC algorithm outperforms the FCM algorithm in most cases. Delgado et al. (1997) reported a tabu search fuzzy clustering algorithm and adopted the probability threshold to generate neighboring solutions by moving cluster centroids or shaking memberships. The aforementioned methods (Al-sultan and Fedjki, 1997; Delgado et al., 1997) generated the trial solutions by moving the coordinates of cluster centers in random. But how to fine-tune the membership degrees of objects with respect to different clusters so as to improve the object

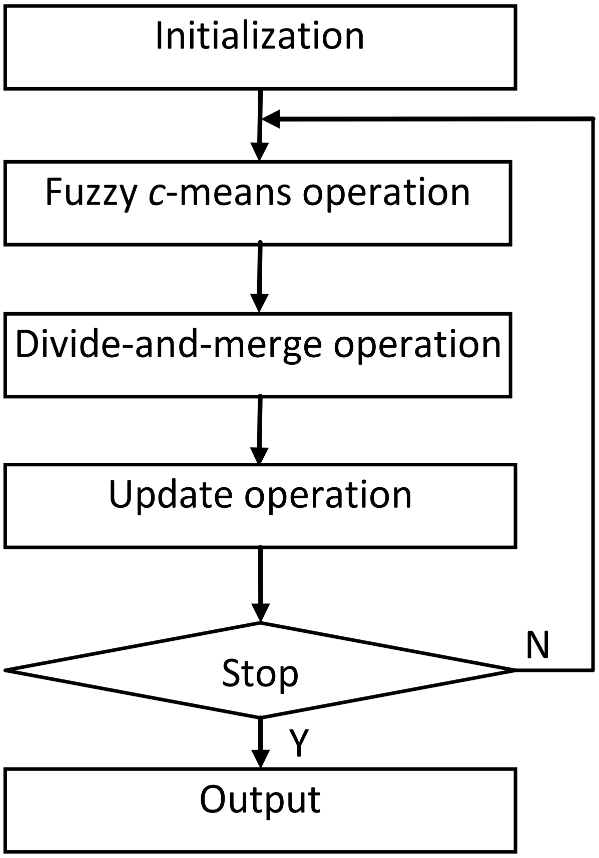


Figure 1. General description of the ITSFC algorithm.

distribution among different clusters did not receive enough attention in their works. After reviewing the related work, we find that many research works focus on employing tabu search to solve the hard clustering problem (Al-sultan, 1995; Liu et al., 2008; Sung and Jin, 2000), but relatively few attempts have been made to solve the fuzzy clustering problem with tabu search. So, it is necessary to further improve the performance of the tabu search fuzzy clustering method.

Our aim is to develop an improved tabu search fuzzy clustering algorithm and demonstrate the effectiveness of the ITSFC algorithm for the fuzzy clustering problem under consideration.

MATERIALS AND METHODS

The ITSFC algorithm

The ITSFC algorithm observes the architecture of tabu search, integrates a one-step fuzzy *c*-means algorithm as the fuzzy *c*-means operation to improve the current solution and accelerate the convergence speed of the clustering method, and designs the divide-and-merge operation to modulate the object distribution among different clusters and create the set of neighboring solutions. The general description of the ITSFC algorithm is shown as Figure 1. In this study, the clustering solution is made up of real numbers representing the coordinates of cluster centers. Then the length of the solution is $c \times d$, where c is the number of clusters and d is the number of object attributes. The first d elements denote the d dimensions of the first cluster center; the next d elements represent those of the second cluster center and so on. For instance, let $c = 2$ and $d = 2,$ then the solution (2.7, 9.5, 3.8, 1.6) represents the coordinates of two cluster centers [(2.7, 9.5) (3.8, 1.6)]. For generating an initial solution, we randomly choose c distinct objects from the data set and view them as the initial cluster centers. The implementation of the ITSFC algorithm will be stated as follows:

Fuzzy c-means operation

In this paper, we integrate the one-step fuzzy *c*-means algorithm into the ITSFC algorithm in order to incorporate the domain knowledge into the clustering procedure so as to enhance the convergence speed of the clustering algorithm. The fuzzy *c*-means operation is stated as follows: given the current solution X_c, compute the membership degree of the *k*th object to the *i*th cluster:

$$\mu_{ik} = \left[\sum_{j=1}^{C} \left(\frac{\left\| \mathbf{x}_k - \mathbf{c}_i \right\|^2}{\left\| \mathbf{x}_k - \mathbf{c}_j \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{2}$$

and then update cluster center $\mathbf{c}_i$

$$\mathbf{c}_i = \frac{\sum_{k=1}^{N} [\mu_{ik}]^m \mathbf{x}_k}{\sum_{k=1}^{N} [\mu_{ik}]^m} \quad (3)$$

Where, $i, j = 1, \ldots, C$, $i \neq j$ and $k = 1, \ldots, N$.

After the membership function values of all objects and cluster centers are updated, the improved solution is viewed as the current solution X_c.

Divide-and-merge operation

In this study, we design the divide-and-merge operation to modulate the object distribution among different clusters, establish the set of neighboring solutions and to update the current solution X_c. The divide-and-merge operation is composed of two modes: division mode and merger mode. On one hand, the division mode is employed to choose the cluster to be partitioned and divide this cluster into two new clusters, and on the other hand, the merger mode is designed to select the cluster to be absorbed and reassign the objects belonging to the merged cluster among the remaining clusters. These two modes are performed in a random order to create a neighboring solution. The divide-and-merge operation is stated as follows:

Step 1: Given the current solution X_c and the number of neighboring solutions N_t, set the counter of the neighborhood $i = 1$.

Step 2: The divide-and-merge operation is performed as follows:

1) If solution X_c is assigned into the division mode, we employ the 2-fold tournament selection, a selection operation in the genetic algorithm to determine the cluster to be divided. Firstly, two clusters C_j and C_k, $j \neq k$, are randomly chosen and the cluster with the sparsest structure is viewed as the candidate cluster $C_{division}$, that is, cluster $C_{division}$ should have the maximum average objective function value. Then two objects x_p and x_q, $p \neq q$, belonging to cluster $C_{division}$ are randomly chosen as the centers of two new clusters *C'* and *C"*. After the division mode, the cluster center of cluster $C_{division}$ is deleted and the number of clusters increases by one. Finally, the membership degree of each object with respect to each cluster is updated.

2) If solution X_c is assigned into the merger mode, like the division mode, we employ the 2-fold tournament selection to choose the cluster to be absorbed. Firstly, two cluster center pairs are randomly selected, the cluster center pair that is the closest in Euclidean distance is determined, and then the cluster of this pair with the sparsest structure is viewed as the cluster to be merged C_{merger}. If there are only two clusters, the cluster with the sparsest structure is then defined as cluster C_{merger}. That is, cluster C_{merger} has the maximum average objective function value. As a result, the cluster center of cluster C_{merger} is deleted and the number of clusters decreases by one. Finally, the membership degrees of objects with respect to the remaining clusters are updated. After the divide-and-merge operation, the number of clusters is kept fixed and the *i*th neighboring solution X_i is established.

Step 3: If $i \leq N_t$, then $i = i + 1$; and go to Step 2 otherwise return the set of neighboring solutions.

Implementation of the ITSFC algorithm

The ITSFC algorithm is implemented as follows:

Step 1: Initialization

Generate an initial solution X_0 at random, and let $X_c = X_b = X_0$ and $J_c = J_b = J(X_0)$, where J_c denotes the objective function value of the current solution, X_c and J_b denotes the objective function value of the best known solution X_b. Set the counter of iterations $i = 1$.

Step 2: Fuzzy c-means operation

Tune solution X_c by the fuzzy *c*-means operation. If $J_c < J_b$, then let $J_b = J_c$ and $X_b = X_c$.

Step 3: Creation of the neighborhood

Generate a neighboring solution X_j by the divide-and-merge operation and compute its objective function value $J(X_j)$, where $j = 1, \ldots, N_t$. Order $J(X_1), \ldots, J(X_{N_t})$ in an ascending order and denote them by $J(X_1'), \ldots, J(X_{N_t}')$, then the ordered neighboring solutions are denoted by $X_1', \ldots, X_{N_t}'$.

Step 4: Update of the current solution

If the neighboring solution X_1' is not tabu or it is tabu but satisfies the aspiration criterion, that is $J(X_1') < J_b$, then let $X_c = X_1'$ and $J_c = J(X_1')$, and proceed to Step 5; otherwise let $X_c = X_l'$, $J_c = J(X_l')$ proceed to Step 5, where X_l' denotes the neighboring solution that is not tabu and has the minimum objective function value among the remaining ordered neighboring solutions and $J(X_l')$ denotes the objective function value of solution X_l'. If all neighboring solutions are tabu, then go to Step 2.

Step 5: Update of tabu list

Add solution X_c at the bottom of the tabu list and set $t = t + 1$, where t denotes the counter of the tabu list. If $t > T$, then remove the first item and set $t = t - 1$, where T denotes the size of the tabu list. If $J_c < J_b$, then let $J_b = J_c$ and $X_b = X_c$.

Step 6: Termination check

If $i \le G$, then $i = i+1$ and go to Step 2; otherwise output solution X_b, where *G* denotes the number of iterations.

RESULTS

Here, computer simulations are conducted in Matlab on an Intel Core 2 Duo processor running at 3 GHz with 4 GB real memory. Each experiment includes 20 independent trials. As the FCM algorithm is inferior to the methods reported in Al-sultan and Fedjki (1997) and Karaboga and Ozturk (2010), we here focus on comparing the ITSFC algorithm with the TSFC algorithm and the ABC-FC algorithm for two artificial and four real life data sets. All real life data sets are available at http://ftp.ics.uci.edu/pub/machine-learning-databases/. Experimental data sets are described as follows:

i) Data-52 (*N* = 250, *d* = 2, *c* = 5) consists of 250 overlapping objects where the number of clusters is five (Liu et al., 2008).
ii) Data-62 (*N* = 300, *d* = 2, *c* = 6) consists of 300 non-overlapping objects where the number of clusters is six (Liu et al., 2008).
iii) Fisher's iris data set (*N* = 150, *d* = 4, *c* = 3) which consists of three different species of iris flower: *Iris setosa*, *Iris virginica* and *Iris versicolour*. For each species, 50 samples with four features each (sepal length, sepal width, petal length and petal width) are collected.
iv) Wine (*N* = 178, *d* = 13, *c* = 3) consists of 178 objects characterized by 13 such features as alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and praline are the results of a chemical analysis of wines brewed in the same region in Italy but derived from three different cultivars.
v) Ripley's glass data set (*N* = 214, *d* = 9, *c* = 6), which consists of six different types of glass: building windows float processed, building windows non-float processed, vehicle windows float processed, containers, tableware and headlamps each with 9 features which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron.
Vi) Contraceptive method choice data set (*N* = 1473, *d* = 9, *c* = 3), which consists of a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who either were not pregnant or did not know if they were at the time of interview. The problem is to predict the choice of current contraceptive method (no use has 629 objects, long-term methods have 334 objects and short-term methods have 510 objects) of a woman based on her demographic and socioeconomic characteristics.

The settings of parameters are described as follows: in the TSFC algorithm, the probability threshold *P* is equal to 0.97, the direction multiplier *α* is equal to 1 and reduced every time by a factor of 0.8, the maximum number of iterations for each center is equal to 10, the iteration reducer *β* is equal to 0.75, the size of the tabu list is equal to 20 and the neighborhood size is equal to 20. In the ABC-FC algorithm, the colony size is set to 20, the limit value is set to 30 and the number of iterations is set to 200. Detailed descriptions of these parameters can be found in their corresponding references. In the ITSFC algorithm, for a fair performance comparison, the size of the tabu list and the neighborhood size are the same as those in the TSFC algorithm and the number of iterations is the same as that in the ABC-FC algorithm. The average (Avg) and standard deviation (SD) values of the objective function are shown in Table 1. Among experimental methods, the ITSFC algorithm outperforms the other two methods and provides the minimum average values for all experimental data sets. In addition, the standard deviation value for each data set reported by the ITSFC algorithm is far less than those given by the TSFC algorithm and the ABC-FC algorithm. In order to understand the performance of three metaheuristic clustering methods better, we show the iteration process for each data set in Figure 2.

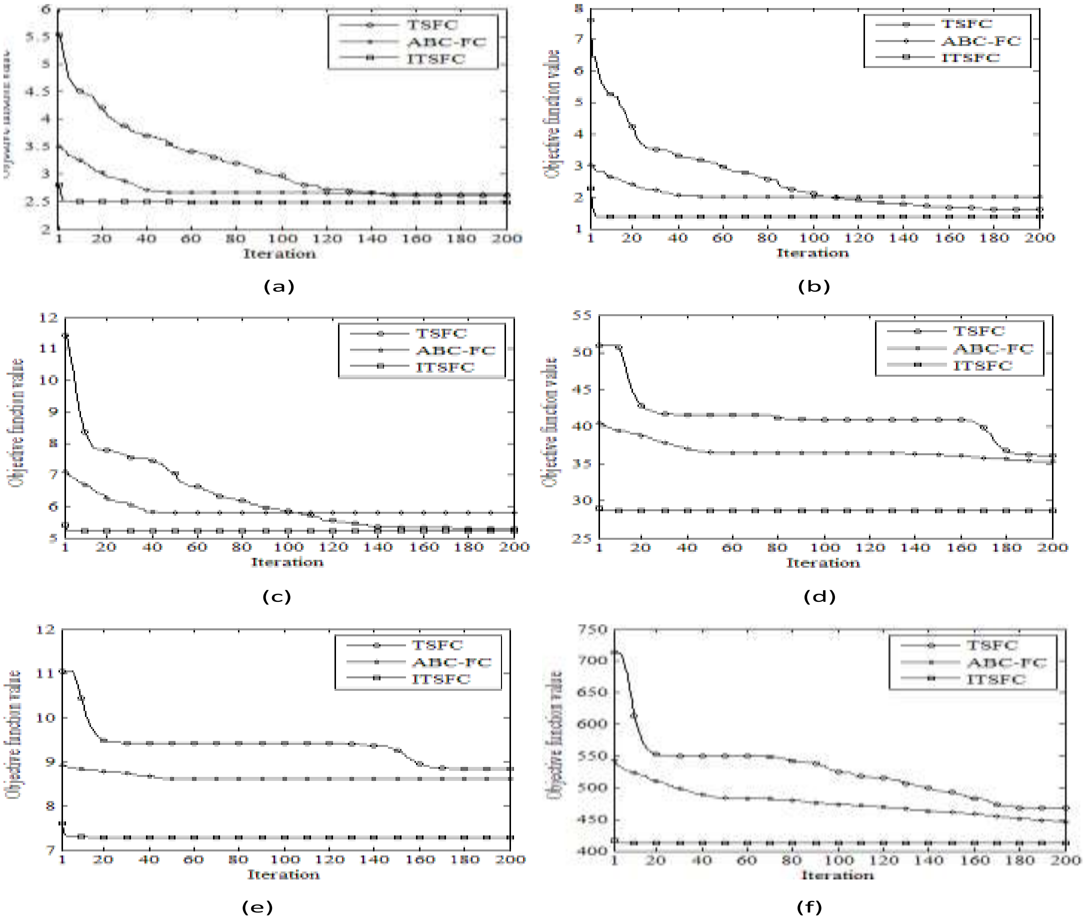
It is seen that the convergence speed of the ABC-FC algorithm is faster than that of the TSFC algorithm at the beginning of iterations, but with the increase of the number of iterations, the latter converges sooner than the former and even provides better results for Data-52, Data-62 and Iris. In face of all data sets, the ITSFC algorithm finds the better results within less number of iterations than the other two methods. The average (Avg) and standard deviation (SD) values of run time when the minimum function values are firstly attained by different methods are recorded in Table 2. In all experiments, we find that the ABC-FC algorithm takes less run time than the other two methods to find its minimum objective function values but cannot output meaningful clustering results. In face of two artificial data sets, the ITSFC algorithm requires more run time than the TSFC algorithm to output better clustering results. Considering four real life data sets with the increase of the number of objects, the ITSFC algorithm provides lower objective function values sooner than the TSFC algorithm.

DISCUSSION

In this study, we find that the ITSFC algorithm takes more run time to achieve better clustering results than the TSFC algorithm for two artificial data sets. Meanwhile, in face of four real life data sets, the proposed algorithm outputs lower objective function values much sooner than the TSFC algorithm. So, to further enhance the convergence speed of the ITSFC algorithm by keeping a good balance between the fuzzy *c*-means operation and the divide-and-merge operation will be our focus of attention in future

Table 1. Clustering results of different experimental methods.

	TSFC Avg SD	ABC-FC Avg SD	ITSFC Avg SD
Data-52	2.6028 0.1050	2.6404 0.0662	2.4972 3.8×10^{-4}
Data-62	1.6199 0.1725	2.0184 0.2826	1.3769 3.8×10^{-7}
Iris	5.2629 0.0580	5.7989 0.2756	5.2331 3.1×10^{-5}
Wine	30.1152 0.6550	35.3164 0.9902	28.7165 7.8×10^{-5}
Glass	7.3944 0.0415	8.6184 0.2438	7.2902 2.6×10^{-4}
Contraceptive method choice	413.4182 0.7587	445.7260 9.8918	412.7560 0.0146



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2. Comparison of three clustering methods on experimental data sets; a) Data-52, b) Data-62, c) Iris, d) wine, e) glass and (f) contraceptive method choice.

Table 2. Run time of different experimental methods.

	TSFC Avg SD	ABC-FC Avg SD	ITSFC Avg SD
Data-52	26.11	21.17	48.21
	7.20	13.54	25.85
Data-62	41.89	12.08	76.49
	9.19	2.60	34.74
Iris	23.18	3.79	20.41
	9.81	2.23	11.25
Wine	154.39	15.00	24.94
	56.29	6.76	12.80
Glass	592.45	10.09	43.45
	163.49	1.59	31.09
Contraceptive method choice	1561.15	152.75	191.59
	392.39	13.93	117.67

research. In addition, as the proposed method requires the designer to provide the number of clusters as input, but in many real-life cases the number of clusters in a data set is not known ‘a priori’, in this case, how to evolve the number of clusters automatically, how to establish neighboring solutions with different numbers of clusters and how to develop accurate criteria to quantitatively measure the quality of the fuzzy partition obtained will be the other subjects of our future research.

Conclusion

In real applications there are often no sharp boundaries between clusters so that data objects might partially belong to multiple clusters. Under this condition, fuzzy clustering instead of hard clustering becomes a good choice for grouping data sets. As the FCM algorithm is sensitive to initialization and may be trapped into local optima, researcher recently employed metaheuristic techniques to solve the fuzzy clustering problem. In this paper, the ITSFC algorithm is proposed in which the fuzzy *c*-means operation is adopted to accelerate the convergence speed of the clustering algorithm and the divide-and-merge operation is developed to establish neighboring solutions. Experimental results on two artificial and four real life data sets are conducted to show that the ITSFC algorithm can provide better objective function values than the other two fuzzy clustering methods.

ACKNOWLEDGMENTS

This research was supported in part by the National Natural Science Foundation of China (NSFC) under grants 60903074 and 60828005, the National High Technology Research and Development Program of China (863 Program) under grant 2008AA01Z119, the Open Project Program of Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education under grant 2009ICIP01, the National Basic Research Program of China (973 Program) under grant 2009CB326203 and the US National Science Foundation (NSF) under grant CCF-0905337.

REFERENCES

Al-sultan KS (1995). A tabu search approach to the clustering problem. Pattern Recog., 28(9): 1443-1451.

Al-sultan KS, Fedjki CA (1997). A tabu search-based algorithm for the fuzzy clustering problem. Pattern Recog., 30(12): 2023-2030.

Amiri B, Fathian M, Maroosi A (2009). Application of shuffled frog-leaping algorithm on clustering. Int. J. Adv. Man. Technol., 45(1-2): 199-209.

Baraldi A, Blonda P (1999). A survey of fuzzy clustering algorithms for pattern recognition. IEEE Trans. Syst. Man Cybern. Part B, Cybern., 29(6): 778-785.

Bezdek JC (1981). Pattern recognition with fuzzy objective function algorithms. Plenum, New York.

Chang DX, Zhang XD, Zheng CW (2009). A genetic algorithm with gene rearrangement for *k*-means clustering. Pattern Recog., 42(7): 1210-1222.

Chen W, Lu TJ, Wu JZ, Zhao YQ (2010). An improved genetic FCM clustering algorithm. In: Proceedings of International Conference on Future Computer and Communication, Wuhan, China, pp. 45-48.

Delgado M, Skármeta AG, Barberá HM (1997). A tabu search approach to the fuzzy clustering problem. In: Proceedings of IEEE International Conference on Fuzzy Systems, Barcelona, Spain, pp. 125-130.

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2010). A review of robust clustering methods. Adv. Data Anal. Classif, 4(2-3):

89-109.

Gendreau M, Potvin JY (2010). Handbook of Metaheuristics. Springer, New York.

Glover F, Laguna M (1997). Tabu search. Kluwer Academic Publishers, Boston.

Izakian H, Abraham A (2011). Fuzzy *c*-means and fuzzy swarm for fuzzy clustering problem. Expert Syst. Appl., 38(3): 1835-1838.

Jarhoui B, Cheikh M, Siarry P, Rebai A (2007). Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Appl. Math. Comput., 192(5): 337-345.

Kanade PM, Hall LO (2007). Fuzzy ants and clustering. IEEE Trans. Syst. Man Cybern. Part A, Syst. Hum., 37(5): 758-769.

Karaboga D, Akaya B (2009). A comparative study of artificial bee colony algorithm. Appl. Math. Comput., 124(1): 108-132.

Karaboga D, Ozturk C (2010). Fuzzy clustering with artificial bee colony algorithm. Sci. Res. Essays., 5(14): 1899-1902.

Laszlo M, Mukherjee S (2006). A genetic algorithm using hyper-quadtrees for low-dimensional *k*-means clustering. IEEE Trans. Pattern Anal. Mach. Intell., 28(4): 533-543.

Liu YG, Yi Z, Wu H, Ye M, Chen KF (2008). A tabu search approach for the minimum sum-of-squares clustering problem. Inf. Sci., 178(12): 2680-2704.

Mehdizadeh E, Sadi-Nezhad S, Tavakkoli-Moghaddam R (2008). Optimization of fuzzy clustering criteria by a hybrid PSO and fuzzy *c*-means clustering algorithm. Iran. J. Fuzzy. Syst., 5(3): 1-14.

Omran MGH, Engelbrecht AP, Salman A (2007). An overview of clustering methods. Intell. Data Anal., 11(6): 583-605.

Pedrycz W (2005). Knowledge-based clustering. Wiley, New Jersey.

Selim SZ, Ismail MA (1984). K-means-type algorithm: generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell., 6(1): 81-87.

Sung CS, Jin HW (2000). A tabu-search-based heuristic for clustering. Pattern Recognit. 33(5): 849-858.

Supratid S, Kim H (2009). Modified fuzzy ants clustering approach. Appl. Intell., 31(2): 122-134.

Xu R, Wunsch D (2008). Clustering. Wiley, New Jersey.

Zhang CS, Ouyang DT, Ning JX (2010). An artificial bee colony approach for clustering. Expert Syst. Appl., 37(7): 4761-4767.