Finding the Optimal Number of Clusters Using Genetic Algorithms

Yongguo Liu*†, Mao Ye*, Jun Peng‡ and Hong Wu*
*School of Computer Science and Engineering
University of Electronic Science and Technology of China, Chengdu 610054, P. R. China
†National Laboratory on Machine Perception, Peking University, Beijing 100871, P. R. China
‡College of Electronic Information Engineering
Chongqing University of Science and Technology, Chongqing 400050, P. R. China

***Abstract*—In clustering analysis, many methods require the designer to provide the number of clusters. Unfortunately, the designer has no idea, in general, about this information beforehand. In this paper, we propose a genetic algorithm based clustering method called Automatic Genetic Clustering for Unknown K (AGCUK). The AGCUK algorithm is able to automatically provide the number of clusters and find the clustering partition. The Davies-Bouldin index is employed to measure the validity of the clusters. Experimental results on artificial and real-life data sets are given to illustrate the effectiveness of the AGCUK algorithm.**

***Index Terms*—clustering, genetic algorithms, noising method, Davies-Bouldin index.**

I. Introduction

Clustering is a formal study of algorithms and methods for classifying objects without category labels. A cluster is a set of entities that are alike, and entities from different clusters are not alike. Many clustering techniques have been proposed [1], [2]. Among them, the K-means algorithm is an important one. However, it often gets stuck at local minima and its result is largely dependent on the choice of the initial cluster centers [3]. Recently, researchers solved the clustering problem by the metaheuristic algorithms. Liu et al. [4] integrated a tabu list into the genetic algorithm based clustering algorithm to prevent several fitter individuals from occupying the population and to maintain population diversity. In addition, an aspiration criterion is adopted to keep selection pressure. Bandyopadhyay and Maulik [5] designed a genetic clustering approach. They employed the K-means algorithm to provide the domain knowledge and improve the search capability of genetic algorithms. Liu et al. [6] combined the K-means algorithm and the tabu search approach to accelerate the convergence speed of the tabu search based clustering algorithm. Ng and Wong [7] applied the tabu search approach to the fuzzy clustering problem so as to deal with categorical data sets. Bandyopadhyay et al. [8] integrated the K-means algorithm into the simulated annealing based clustering method to improve the cluster centroids. By redistributing objects among clusters probabilistically, the presented method obtains better results than the K-means algorithm. Liu et al. [9] adopted the noising method, a metaheuristic technique proposed by Charon and Hudry [10], to solve the clustering problem. The proposed method requires less computational cost than Bandyopadhyay et al.'s algorithm [8] but is still inferior to the latter. The aforementioned methods [3]–[9] require the designer to provide the number of clusters. Unfortunately, in many real-life cases the number of clusters in a data set is not known a priori. Under this condition, how to automatically provide the number of clusters and find the clustering partition becomes a challenge.

In this regard, some attempts have been made to use genetic algorithms for automatically clustering data sets. Bandyopadhyay and Maulik [11] applied the variable string length genetic algorithm with the real encoding of the coordinates of the cluster centers in the chromosome to the clustering problem. Experimental results show that their algorithm is able to evolve the number of clusters as well as provide the proper clustering. Tseng and Yang [12] proposed a genetic algorithm based approach for the clustering problem. Their method consists of two stages, nearest neighbor clustering and genetic optimization. The proposed method can search for a proper number of clusters and classify nonoverlapping objects into these clusters. Bandyopadhyay and Maulik [13] exploited the searching capability of genetic algorithms for providing the number of clusters of a given data set. A string representation composed of real numbers and the do not care symbol is used to encode a variable number of clusters. Effectiveness of their technique is demonstrated for experimental data sets. Lin et al. [14] presented a genetic clustering algorithm based on a binary chromosome representation. The proposed method selects the cluster centers directly from the data set. With the aid of a look-up table, the distances between all pairs of objects are saved in advance and evaluated only once throughout the evolution process. By experimental simulations, the superiority of their algorithm over Bandyopadhyay and Maulik's method [13] is shown. Lai [15] adopted the hierarchical genetic algorithm to solve the clustering problem. In the proposed method, the chromosome consists of two types of genes, control genes and parametric genes. The control genes are coded as binary digits. The total number of "1" represents the number of clusters. The parametric genes are coded as real numbers to represent the coordinates of the cluster centers. The relationship between the control genes and the parametric genes is that the activation of the latter is governed by the value of the former. If the value of a control gene is "1", then

978–1–4244–1674–5/08/$25.00 © 2008 IEEE

the associated parametric genes due to that particular active control gene are activated; otherwise the associated parametric genes are disabled. Experimental results show Lai's method can search for the number of clusters.

In this paper, our aim is to develop a new genetic algorithm based clustering method to automatically provide the number of clusters and find the clustering partition. We design two operators, noising selection and division-absorption mutation. The Davies-Bouldin index is employed as a measure of the validity of the clusters. As a result, a new genetic clustering method called Automatic Genetic Clustering for Unknown K (AGCUK) is proposed. Experimental results on artificial and real-life data sets are given to illustrate the superiority of the AGCUK algorithm over some known genetic clustering methods. The remaining part of this paper is organized as follows: In Section II, the AGCUK algorithm and its components are described. Results of computer simulations are given in Section III. Finally, some conclusions are drawn in Section IV.

II. The Proposed Approach

In this section, we first briefly introduce genetic algorithms and the noising method, and then describe the AGCUK algorithm in detail.

A. Genetic Algorithms and Noising Method

Genetic algorithms are randomized search and optimization techniques guided by the principles of evolution and natural genetics [16]. They have a large amount of implicit parallelism and provide the near optimal solution of the objective or fitness function in complex, large and multi-modal landscapes. In genetic algorithms, the parameters of the search space are encoded in the form of strings (or chromosomes). The fitness function that represents the goodness degree of the solution encoded in the chromosome is associated with each string. Biologically inspired operators like selection, crossover and mutation are used over a number of generations for generating potentially better solutions. After a satisfactory individual is found or the specified number of generations is over, the best individual obtained is viewed as the final result. Genetic algorithms have applications in fields as diverse as image processing, information security, information retrieval, etc. [17]–[19].

The noising method guiding the heuristic search procedure to explore the solution space is a metaheuristic technique proposed by Charon and Hudry [10]. Instead of taking the genuine data into account directly, the noising method considers the optimal result as the outcome of a series of fluctuating data converging towards the genuine ones. Like some other metaheuristics, the noising method is based on a descent. The main difference with a descent is that, when the objective function value for a given solution is considered, a perturbation called a noise is added to this value. This noise is randomly chosen in an interval of which the range decreases during the iteration process. It means that the original value of the noise rate r_n should be chosen in such a way that, at the beginning of the iteration process, a bad neighboring solution may be accepted. As added noises are chosen in an interval centered on zero, a good neighboring solution may be also rejected. The final solution is the best solution obtained during the iteration process. The noising method is described as Fig. 1. In Fig. 1, N_i denotes the current number of iterations, N_t denotes the total number of iterations, N_f denotes the number of iterations at a fixed noise rate, $f(X_c)$, $f(X')$ and $f(X_b)$ denote the function values of the current solution X_c, the neighboring solution X' and the best known solution X_b, respectively. The noising method has been applied to traveling salesman problem [20], task allocation problem [21], and clique partitioning problem [22], etc.

```
Begin
    set parameters and create the current solution $X_c$
    while $N_i \leq N_t$ do
        $N_i \leftarrow N_i + 1$
        find the neighboring solution $X'$ of solution $X_c$
        if $f(X') - f(X_c) + noise < 0$, then $X_c \leftarrow X'$
        if $f(X_c) < f(X_b)$, then $X_b \leftarrow X_c$
        if $N_i = 0(\mathrm{mod} N_f)$, then decrease $r_n$
    end do
end
```

Fig. 1. The structure of the noising method.

B. The AGCUK Algorithm

The AGCUK algorithm is described as shown in Fig. 2. Its most procedures are based on the architecture of genetic algorithms. The following subsections consider the design approaches in great detail.

```
Begin
    initialize parameters
    establish the initial population for clustering
    while (not termination condition) do
        fitness evaluation
        noising selection
        division-absorption mutation
    end do
end
```

Fig. 2. The general description of the AGCUK algorithm.

1) Individual representation: In the AGCUK method, the chromosomes are made up of real numbers to represent the coordinates of the cluster centers. The length of the chromosome is $K_i * m$, where K_i denotes the number of clusters of the i^{th} individual and m denotes the number of object attributes. The first m genes denote the m dimensions of the first cluster center, the next m genes represent those of the second cluster center, and so on. For instance, let $m = 2$ and $K_i = 3$, then the individual $\{25.2\ 18.6\ 5.3\ 10.8\ 65.3\ 7.0\}$ represents the coordinates of three cluster centers $\{(25.2\ 18.6)(5.3\ 10.8)(65.3\ 7.0)\}$.

2) Population initialization: For individual i, its number of clusters K_i is randomly generated in the range $[K_{min}, K_{max}]$. Here, K_{min} is chosen to be 2 unless specified otherwise and K_{max} is chosen to be $\sqrt{N}$, where N denotes the number of objects. For initializing individual i, K_i distinct objects are chosen randomly from the data set and viewed as the initial cluster centers.

3) Fitness evaluation: The aim of clustering analysis is to divide a given data set into clusters. A resulting partition should possess the following properties: (1) homogeneity within the clusters, i.e. data that belong to the same cluster should be as similar as possible, and (2) heterogeneity between the clusters, i.e. data that belong to different clusters should be as different as possible. In this paper, we use the Davies-Bouldin (DB) index [23] to compute the fitness of the individual. The DB index is also adopted in [11], [13]–[15] to measure the validity of the clusters. The DB index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within cluster C_i is defined as

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|_2^q \right)^{\frac{1}{q}}, \tag{1}$$

where $\mathbf{c}_i$ denotes the cluster center of cluster C_i and $S_{i,q}$ denotes the q^{th} root of the q^{th} moment of the objects belonging to cluster C_i with respect to their mean. $S_{i,q}$ is a measure of the dispersion of the objects belonging to cluster C_i. Specifically, $S_{i,1}$, used in this article, denotes the average Euclidean distance of the objects belonging to cluster C_i to their cluster center $\mathbf{c}_i$. Cluster center $\mathbf{c}_i$ is computed as

$$\mathbf{c}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}, \tag{2}$$

where n_i denotes the number of the objects belonging to cluster C_i. The distance between clusters C_i and C_j is defined as

$$d_{ij,t} = d(C_i, C_j) = \|\mathbf{c}_i - \mathbf{c}_j\|_t, \tag{3}$$

where $d_{ij,t}$ denotes the Minkowski distance of order t between the centroids which characterize clusters C_i and C_j. Next, we define a term $R_{i,qt}$ for cluster C_i as

$$R_{i,qt} = \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}. \tag{4}$$

Then the DB index is defined as

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt}. \tag{5}$$

A small value of this evaluation indicates a good clustering result, and thus we set the fitness F_i of individual i to be equal to $1/DB_i$, where DB_i is the DB index computed for individual i. Then the minimum problem is converted into a maximum one suitable for genetic algorithms.

4) Noising selection: We adopt the noising method to implement the selection operation so as to avoid the selected population being occupied by several fitter individuals and to maintain population diversity. We add noises to the variation of the fitness value. The selection operation is implemented as follows:

Step 1: Given population Q^t, where t denotes the number of generations, set $i = 1$ and choose the i^{th} individual X_i^t.

Step 2: If $t = 1$, then individual X_i^t is selected and proceed to Step 4.

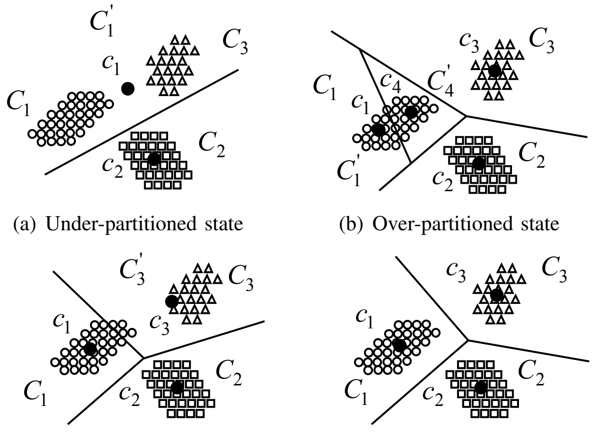
Step 3: Individual X_i^t is compared with the i^{th} individual X_i^{t-1} in population Q^{t-1}, if

$$F_i^t - F_i^{t-1} + noise > 0, \tag{6}$$

then individual X_i^t is selected; otherwise individual X_i^{t-1} is selected. Here, F_i^t and F_i^{t-1} denote the fitness values of individuals X_i^t and X_i^{t-1}, respectively.

Step 4: View the selected individual as the i^{th} individual and let $i = i + 1$. If $i \leq P$, then return to Step 2; otherwise output the selected population. Here, P denotes the population size.

5) Division-absorption mutation: There are three partition states, under-partitioned state, optimal-partitioned state and over-partitioned state, for a given data set as shown in Fig. 3. In the under-partitioned state, two clusters C_1 and C_3 are improperly grouped into cluster $C_1^{'}$. In the over-partitioned state, cluster C_1 is improperly divided into clusters $C_1^{'}$ and $C_4^{'}$. Only in the optimal-partitioned one, all clusters are correctly divided. So, partitioning the under-partitioned cluster and merging the over-partitioned cluster are helpful for exploring the correct clustering. In this paper, we design the division-absorption mutation composed of two sub-operations, division operation and absorption operation. In the AGCUK approach, there are two kinds of individuals, the best individuals and the others. The best individuals have the highest fitness. Here, we view the best individuals as the solutions with the "correct" number of clusters. But they may not represent the optimal partition



(a) Under-partitioned state

(b) Over-partitioned state

(c) Suboptimal-partitioned state

(d) Optimal-partitioned state

Fig. 3. Partition of data sets.

as shown in Fig. 3(c). For these individuals, we keep the number of clusters constant and perform two sub-operations in random order to improve the assignment of objects and find the clustering partition. For the other individuals, we randomly choose one sub-operation to redistribute objects among clusters so as to explore the number of clusters. Two sub-operations are stated as follow:

Division operation: Suppose cluster C_i is the one to be divided, we use proportional selection to choose cluster C_i. The selection probability is defined as

$$p_i = S_{i,1} / \sum S_{i,1}, \tag{7}$$

where $i = 1, \ldots, K$. That is, the sparser cluster C_i, the more possibly it is selected as the cluster to be divided, and vice versa. Since the K-means algorithm is simple and computationally attractive, we adopt it to partition cluster C_i. After the division operation, cluster C_i is divided into two new clusters and the number of clusters increases by one.

Absorption operation: Like the division operation, we adopt proportional selection to determine which cluster is to be absorbed. That is, the closer two clusters to each other, the more possibly one of them is selected as the one to be absorbed, and vice versa. The distance between cluster C_i and its nearest neighbor is computed as

$$d_i = \min_{i \neq j} \|\mathbf{c}_i - \mathbf{c}_j\|^2, \tag{8}$$

then the selection probability is defined as

$$p_i = \sum d_i / d_i. \tag{9}$$

Suppose cluster pair (C_i, C_j) is selected, if $S_{i,1} > S_{j,1}$ then cluster C_i is selected; otherwise cluster C_j is selected. That is, in the cluster pair, the cluster with sparser structure is the one to be merged. Suppose cluster C_i is to be absorbed, object **x** belonging to cluster C_i is reassigned to cluster C_k, iff

$$\|\mathbf{x} - \mathbf{c}_k\|^2 < \|\mathbf{x} - \mathbf{c}_j\|^2, \tag{10}$$

where $\mathbf{c}_j \neq \mathbf{c}_k$. After the absorption operation, cluster C_i disappears and the number of clusters decreases by one. After the division-absorption mutation, objects are redistributed among clusters and a new individual is created.

6) Termination criterion: In general, two stopping criteria are used in genetic algorithms. In the first, the evolution process is executed for a fixed number of generations and the best individual obtained is taken to be the optimal one. In the other, the algorithm is terminated if no further improvement in the fitness value of the best individual is observed for a fixed number of generations, and the best individual obtained is taken to be the optimal one. We adopt the first method in the experiment. That is, the best individual having the highest fitness seen up to the last generation provides the solution to the clustering problem. In addition, the elitist model is used to carry the best individual obtained from the previous population into the child population, which assures the evolution process to converge towards the optimal result [24].

III. Experimental Results

We compare the AGCUK algorithm with four genetic clustering techniques proposed by Bandyopadhyay and Maulik [11], [13], Lin et al. [14] and Lai [15], respectively. Performance comparisons are conducted in Matlab on an Intel Pentium D processor running at 3.4GHz with 512MB real memory. Each experiment includes 20 independent trials. The settings of parameters are described as follows: The population size P is equal to 20. The number of generations G is equal to 50. In the noising selection operation of the AGCUK algorithm, the original noise rate r_{max} is set to be 10, the terminal noise rate r_{min} is set to be 0, and the number of iterations at the fixed noise rate N_f is set to be 20.

In [14], Lin et al. used 100 artificial data sets with a variety of numbers (in $[K_{min}, K_{max}] = [2, 11]$) of clusters to compare the proposed method with Bandyopadhyay and Maulik's algorithm [13]. There are ten data sets for each number of clusters. As a result, Lin et al.'s method is better than the latter and finds the correct number of clusters and the optimal partitions of the data sets with less than 7 clusters. But Lin et al.'s method becomes bad with the further increase of the number of clusters. In this paper, we use the latter 50 data sets with a variety of numbers (in $[K_{min}, K_{max}] = [7, 11]$) of clusters. Figure 4 shows the sizes of the artificial data sets.

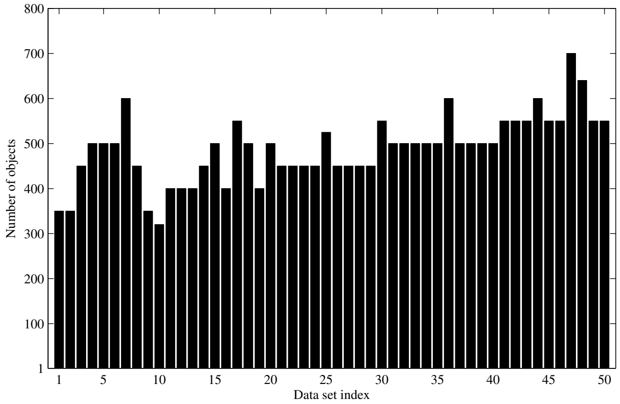


Fig. 4. Sizes of 50 artificial data sets.

Figure 5 shows the average number of clusters provided by experimental algorithms for artificial data sets. It is seen that Bandyopadhyay and Maulik's method [11] is the worst. In face of the data sets with seven clusters, Lai's method is better than the one given by Bandyopadhyay and Maulik [13]. But in face of the other data sets, they are comparable. The number of clusters provided by Lin et al.'s method is more accurate than that provided by above three methods. In face of artificial data sets, the AGCUK algorithm is the best and provides the correct number of clusters in most trials.

In order to compare the ability of experimental methods to find the optimal clustering, we give the average misclassified rates of five clustering methods as shown in Fig. 6. The misclassified rates of the methods reported in [11], [13], [15] are larger than 0 in all runs. Lin et al.'s method and the

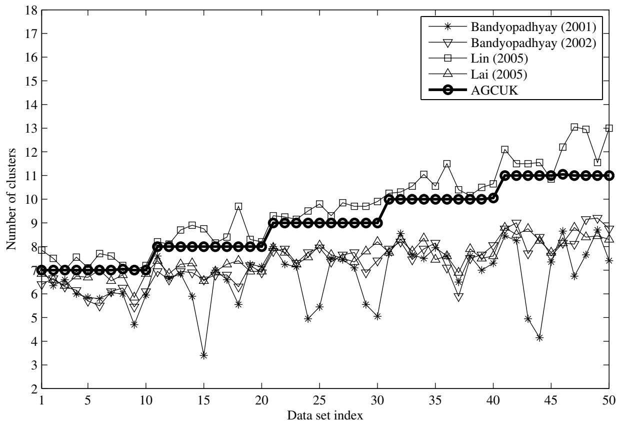


Fig. 5. Number of clusters provided by five methods for artificial data sets.

AGCUK algorithm seem to be comparable. After removing the other three methods, we find that our approach outperforms Lin et al.'s method as shown in Fig. 7. Lin et al.'s method finds the optimal partitions of two data sets (Data 3 and Data 9) in all runs. The AGCUK algorithm provides the optimal partitions of fifteen data sets including Data 3 and Data 9 in each trial.

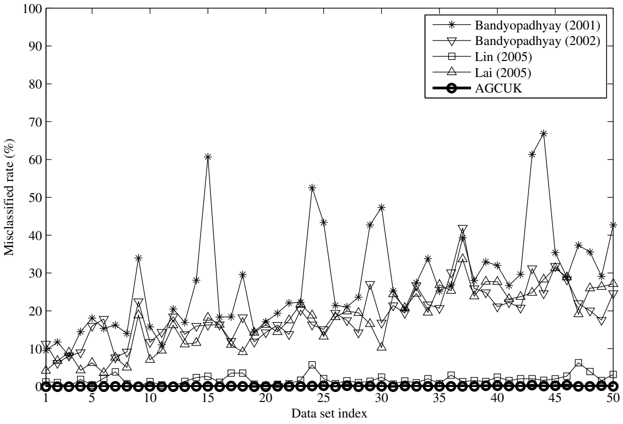


Fig. 6. Misclassified rates of five methods.

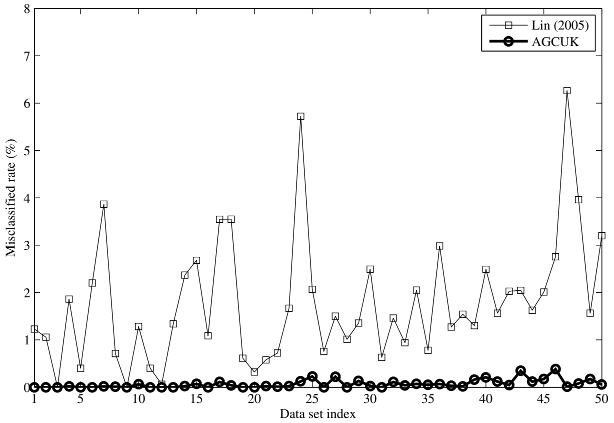


Fig. 7. Misclassified rates of Lin et al.'s method and the AGCUK algorithm.

In addition, we use the Wisconsin Breast Cancer data set [25] to compare experimental algorithms. In Breast Cancer data set, each pattern has nine features corresponding to 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 set, malignant and benign, which are known to be linearly inseparable. The total number of patterns are 699 (458 benign, and 241 malignant), of which 16 patterns contain single missing feature. These 16 patterns have been removed and the remaining 683 patterns are used for clustering.

Experimental results for Breast Cancer are shown as Table I. Bandyopadhyay and Maulik's methods [11], [13] and the AGCUK algorithm provide the correct number of clusters. The misclassified rate of our approach is the lowest. But all experimental algorithms do not provide the optimal partition within the specified number of generations. We find it difficult for one validity index to deal with different data sets. Other validity indices such as PBM-index [26] may be used to find the clustering partition in future research.

TABLE I
RESULTS OF EXPERIMENTAL METHODS FOR BREAST CANCER

	Number of clusters	Misclassified rate (%)
Bandyopadhyay (2001)	2	34.8
Bandyopadhyay (2002)	2	9.0
Lin (2005)	6.2	33.4
Lai (2005)	2.8	15.2
AGCUK	2	3.5

In the following, we analyze the time complexities of experimental methods. The time complexities of the clustering methods reported in [11], [13]–[15] are $O(GPKmN)$, $O(GPKmN)$, $O(GPKN + mN^2)$ and $O(GPKmN)$, respectively. The time complexity of the AGCUK algorithm is given as follows: In each generation, the time complexity of the fitness evaluation is $O(PKmN)$. The time complexity of the selection operation is $O(P)$. The time complexity of the division-absorption mutation is discribed as follows: The time complexity of the division operation is $O(mN)$, and the time complexity of the absorption operation is $O(KmN)$. The time complexity of the division-absorption mutation is dominated by the absorption operation. Therefore, the time complexity of the AGCUK algorithm is $O(GPKmN)$ the same as those of the methods reported in [11], [13], [15].

IV. Conclusions

As a fundamental problem and technique for data analysis, clustering has become increasingly important. Many clustering methods usually require the designer to provide the number of clusters as input. Unfortunately, this information in general is unknown a priori. In this paper, we propose a genetic algorithm based clustering method call Automatic Genetic Clustering for Unknown K (AGCUK). We design two new operations, noising selection and division-absorption mutation. The reciprocal of the Davis-Bouldin index is used for computing the fitness of individuals.

We adopt the noising selection operation to prevent the selected population being occupied by several fitter individuals and to maintain population diversity. Noises are added to the variation of the fitness value so as to avoid the clustering problem being trapped by local minima. According to the clustering partition, we design the division-absorption mutation. Three combinations of division operation and absorption operation are performed on individuals to evolve the number of clusters and find the clustering partition. The AGCUK algorithm and four known genetic clustering techniques are compared. As a result, the AGCUK algorithm can provide the correct number of clusters for artificial and real-life data sets. It obtains lower misclassified rates than the other experimental methods. In addition, the time complexity of the AGCUK method is the same as those of the methods reported in [11], [13], [15].

In this paper, the Davies-Bouldin index is used for computing the fitness of individuals. But we find it difficult to use one validity index to deal with different data sets. Combining other indices such as PBM-index with our method to solve the clustering problem will be an important area of future research.

Acknowledgments

The authors would like to thank Dr. Chih-Chin Lai for his valuable suggestions on our works. This research was supported in part by the Program for New Century Excellent Talents in University (NCET-06-0811) and the Youth Key Foundation of University of Electronic Science and Technology of China.

References

[1] A. K. Jain, M. N. Murty and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[2] W. Pedrycz, *Knowledge-based clustering*, Wiley, 2005.

[3] S. Z. Selim and M. A. Ismail, “K-means-type algorithm: generalized convergence theorem and characterization of local optimality,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 81-87, 1984.

[4] Y. G. Liu, K. F. Chen and X. M. Li, “A hybrid genetic based clustering algorithm,” in *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, IEEE, Shanghai, pp.1677-1682, 2004.

[5] S. Bandyopadhyay and U. Maulik, “An evolutionary technique based on K-means algorithm for optimal clustering in R^N,” *Information Sciences*, vol. 146, no.1-4, pp. 221-237, 2002.

[6] Y. G. Liu, Y. Liu, L. B. Wang and K. F. Chen, “A hybrid tabu search based clustering algorithm,” *Lecture Notes in Computer Science*, vol. 3682, pp. 186-192, 2005.

[7] M. K. Ng and J. C. Wong, “Clustering categorical data sets using tabu search techniques,” *Pattern Recognition*, vol. 35, no. 12, pp. 2783-2790, 2002.

[8] S. Bandyopadhyay, U. Maulik and M. K. Pakhira, “Clustering using simulated annealing with probabilistic redistribution,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 2, pp. 269-285, 2001.

[9] Y. G. Liu, Y. Liu and K. F. Chen, “Clustering with noising method,” *Lecture Notes in Computer Science*, vol. 3584, pp. 209-216, 2005.

[10] I. Charon and O. Hudry, “The noising method: a new method for combinatorial optimization,” *Operations Research Letters*, vol. 14, no. 3, pp. 133-137, 1993.

[11] S. Bandyopadhyay and U. Maulik, “Nonparametric genetic clustering: comparison of validity indices,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31, no. 1, pp. 120-125, 2001.

[12] L. Y. Tseng and S. B. Yang, “A genetic approach to the automatic clustering algorithm,” *Pattern Recognition*, vol. 34, no. 2, pp. 415-424, 2001.

[13] S. Bandyopadhyay and U. Maulik, “Genetic clustering for automatic evolution of clusters and application to image classification,” *Pattern Recognition*, vol. 35, no. 6, pp. 1197-1208, 2002.

[14] H. J. Lin, F. W. Yang and Y. T. Kao, “An efficient GA-based clustering technique,” *Tamkang Journal of Science and Engineering*, vol. 8, no. 2, pp. 113-122, 2005.

[15] C. C. Lai, “A novel clustering approach using hierarchical genetic algorithms,” *Intelligent Automation and Soft Computing*, vol. 11, no. 3, pp. 143-153, 2005.

[16] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, 1989.

[17] P. Kumsawat, K. Attakitmongcol and A. Srikaew, “A new approach for optimization in image watermarking by using genetic algorithms,” *IEEE Trans. on Signal Processing*, vol. 53, no. 12, pp. 4707-4719, 2005.

[18] P. Ramasubramanian and A. Kannan, “A genetic-algorithm based neural network short-term forecasting framework for database intrusion prediction system,” *Soft Computing*, vol. 10, no. 8, pp. 699-714, 2006.

[19] Y. C. Chang and S. M. Chen, “A new query reweighting method for document retrieval based on genetic algorithms,” *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 5, pp. 617-622, 2006.

[20] I. Charon and O. Hudry, “Application of the noising method to the traveling salesman problem,” *European Journal of Operational Research*, vol. 125, no. 2, pp. 266-277, 2000.

[21] W. H. Chen and C. S. Lin, “A hybrid heuristic to solve a task allocation problem,” *Computers & Operations Research*, vol. 27, no. 3, pp. 287-303, 2000.

[22] I. Charon and O. Hudry, “Noising methods for a clique partitioning problem,” *Discrete Applied Mathematics*, vol. 154, no. 5, pp. 754-769, 2006.

[23] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Trans. on Pattern Analysis and Machine Intelligence* vol. 1, no. 2, pp. 224-227, 1979.

[24] D. Bhandari, C. A. Murthy and S. K. Pal, “Genetic algorithm with elitist model and its convergence,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, no. 6, pp. 731-747, 1996.

[25] C. L. Blake and C. J. Merz, “UCI Repository of machine learning databases,” Irvine, CA: University of California, 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html

[26] M. K. Pakhira, S. Bandyopadhyay and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern Recognition*, vol. 37, no. 3, pp. 487-501, 2004.