Soft Comput (2016) 20:1503–1519
DOI 10.1007/s00500-015-1602-z

METHODOLOGIES AND APPLICATION

Co-evolution-based immune clonal algorithm for clustering

Ronghua Shang · Yang Li · Licheng Jiao

Published online: 7 February 2015

Abstract Clustering is an important tool in data mining process. Fuzzy *c*-means is one of the most classic methods. But it has been criticized that it is sensitive to the initial cluster centers and is easy to fall into a local optimum. Not depending on the selection of the initial population, evolutionary algorithm is used to solve the problems existed in original fuzzy *c*-means algorithm. However, evolutionary algorithm emphasizes the competition in the population. But in the real world, the evolution of biological population is not only the result of internal competition, but also the result of mutual competition and cooperation among different populations. Co-evolutionary algorithm is an emerging branch of evolutionary algorithm. It focuses on the internal competition, while on the cooperation among populations. This is more close to the process of natural biological evolution and co-evolutionary algorithm is a more excellent bionic algorithm. An immune clustering algorithm based on co-evolution is proposed in this paper. First, the clonal selection method is used to achieve the competition within population to reconstruct each population. The internal evolution of each population is completed during this process. Second, co-evolution operation is conducted to realize the information exchange among populations. Finally, the iteration results are compared with the global best individuals, with a strategy called elitist preservation, to find out the individual with a highest fitness value, that is, the result of clustering. Compared with four state-of-art algorithms, the experimental results indicate that the proposed algorithm outperforms other algorithms on the test data in the highest accuracy and average accuracy.

Communicated by V. Loia.

R. Shang (✉) · Y. Li · L. Jiao
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China
e-mail: rhshang@mail.xidian.edu.cn

Keywords Immune clone · Co-evolution · Elitist preservation · Clustering · FCM

1 Introduction

Clustering is an important and common technique used to mine the potential characteristics of data. And clustering analysis is a procedure which divides the data into different clusters naturally in the case of little or no priori information. This enables the data within the same cluster to have high similarity and low similarity among different clusters. Clustering algorithm is widely used in data analysis, pattern recognition, image processing and market research (Rao 1971; Lillesand and Keifer 1994). Therefore, cluster analysis has become an increasingly popular area in data mining.

The current major clustering algorithms are as follows: (1) partition-based method: the process of this method first creates an initial partition, then iterates to reposition the individual objects, and ultimately achieves the clustering results by adjusting the range of movement of all of the objects in each division. Typical methods are *k*-means (Dunn 1973), fuzzy *c*-means (Hoppner et al. 1999), etc.; (2) hierarchy-based approach: this method treats all the data objects as a tree. The hierarchical decomposition process from bottom to up is called cohesion clustering. The method regards each object as a separate cluster, and then merges these clusters until all objects are in one cluster. In contrast, the process from up to bottom is called split clustering. This approach is opposite from the cohesion clustering. It regards all the objects as a large cluster and then breaks it down to some small clusters gradually until all the objects are separate clusters. Typical algorithms are BIRCH (Zhang et al. 1996), CURE (Guha et al. 1998), etc.; (3) density-based method: it regards the clusters as high-density regions separated by some low-density regions. The main process is to observe the density (number) around an object point, and when the density exceeds a certain threshold, cluster continues. The advantages of this method are that it has the ability to discover arbitrary shapes clusters and a better noise robustness. Typical algorithms are DBSCAN (Ester et al. 1996), OPTICS (Ankerst et al. 1999), etc.; (4) grid-based method: in this method, the space of the data objects is quantized into a limited number of grids. All the clustering operations are done in each grid. The advantage of the method is fast processing speed, and the processing time is only related to the number of the grid. Typical algorithms are STING (Wang et al. 1997), WaveCluster (Sheikholeslami et al. 1998), CLIQUE (Agrawal et al. 1998), etc.; (5) model-based approach: this approach is to define a model for each cluster and then find out the best fitting of the given model. Typical methods are based on statistical, such as RSDE (Girolami and He 2003) and FRSDE (Deng et al. 2008), and neural network-based methods (Kohonen 1982); (6) spectral graph theory-based method: this method is based on spectral theory. The data objects are defined as an affinity matrix by the similarities between data points, thereby calculate the feature vector, and finally select the appropriate feature vectors to clustering all the data points. Typical algorithms are minimum cuts (Higham and Kibble 2004), normalized cuts (Meila and Xu 2004), etc. In addition, there are some new clustering methods, such as the affinity propagation algorithm (Frey and Dueck 2007; Mézard 2007), the nonnegative matrix factorization-based method (Lee and Seung 1999), a graph-based relaxed clustering (Lee et al. 2008), rough subspace-based clustering ensemble for categorical data (Gao et al. 2013), GACH: a grid-based algorithm for hierarchical clustering of high-dimensional data (Eghbal and Mansoori 2013), improving a multi-objective differential evolution optimizer using fuzzy adaptation and K-medoids clustering (Kotinis 2013), parallel and scalable CAST-based clustering algorithm on GPU (Lin et al. 2013) and cooperative bare-bone particle swarm optimization for data clustering (Jiang and Wang 2013). In all clustering methods, Fuzzy c-means algorithm is one of the most simple and effective, and the most widely used clustering methods. It bases on fuzzy theory, and describes the real world more accurately. Its algorithm is simple and has fast convergence. But it also has its own drawbacks: sensitive to the initial cluster centers and easy to fall into local optimum. This limits its use and accuracy of clustering. But the drawbacks can be solved using the evolutionary computation to some extent.

Evolutionary computation is such an algorithm that simulates the process of biological evolution. It starts searching from multi-points, and it is not easy to fall into local optimum. In addition, the evolutionary algorithm has an adaptive probabilistic searching technology. The selection, crossover and mutation operators are carried out based on a probabilistic approach, thereby increasing the flexibility of its search process. Such evolutionary algorithm does not depend on the selection of the initial population and has a better search capability to find out the global optimal solution. This makes it received widely attentions and applications. These advantages of evolutionary algorithm can be used to solve the problems of fuzzy c-means algorithm exactly.

There is no denying the fact that competition relationship in each species, between species and between species and the environment does exist in nature, but there exist other relationships. With the development of modern biology, it has been found that a variety of organisms are in a certain ecosystem. The existing relationship among the organisms is not just competition, but more importantly is the collaborative relationship. So, although the superiority of evolutionary algorithm has been widely noted, the inferior of most evolutionary algorithms only considering competition within a population, the so-called "survival of the fittest", but not considering the cooperation among populations should not be underestimated. In this case, the co-evolution theory that considering both competition and collaborative relationships begins to get more and more recognition (Jazen 1980).

In 1978, the concept of sub-groups is first mentioned by Holland in his paper. In 1994, co-evolutionary genetic algorithm is proposed by Potter and De Jong (1994). Over a year later, Potter and De Jong have published many papers on the co-evolutionary algorithms (Potter and De Jong 1995, 2000, 1998). Subsequently, Ficici and Watson also studied the co-evolutionary algorithm (Ficici and Pollack 2000; Powers and Watson 2007). Co-evolutionary algorithm is based on the co-evolution theory, which considers both the relationship between two populations, and the relationship between the population and the environment. It is a higher level simulation of biological evolution that overcomes the premature convergence in traditional evolutionary algorithm. The co-evolutionary algorithm is much more close to the process of the biological evolution, and is a more excellent bionic algorithm than the traditional evolution algorithm. Biological evolution is a long process. The common disadvantages of evolutionary algorithm (include the co-evolutionary algorithm) are the slow convergence speed and easy to "premature" and a lot of iterations. However, the immune clonal selection operation has the ability to improve the convergence speed and to select higher fitness individuals in a population while maintaining diversity of the population.

The immune clonal selection theory is an important part of the biological immune system theory. Its principle was first proposed by Jerne, later was fully expanded by Burnet (1957) and has been widely accepted now. The clonal selection algorithm is a process simulating by the immune clonal selection theory. It is self-encoded so that the search process has nothing to do with the problem (de Castro and Zuben 2000; Kim and Bentley 2002). The clonal selection algorithm has many advantages, for example, improving the convergence speed of the algorithm, increasing the diversity of the population while keeping the best individuals. Because of these advantages, a lot of researches have begun to focus on immune clonal selection clustering methods. Chen et al. (2008) combine the immune clonal selection method with classical hierarchical clustering algorithm and proposed a dynamic clonal selection immune clustering algorithm, which could achieve clustering without priori knowledge. Al-Muallim and El-Kouatly (2010) proposed a data-driven and adaptive classification method. Zhong and Zhang (2011, 2012) proposed a new fuzzy clustering method, which has two steps. The first step is used to cluster the data according to the immune clonal selection theory. And the second step is used to adjust the number of clusters to output the best result. Also, Ahmad and Narayanan (2011) discuss a new immune clonal algorithm called population-based artificial immune system clustering algorithm. All of the above researches have been proven to be efficient. However, as a part of artificial immune system, immune clonal selection algorithm overrelies on local research and lacks of learning mechanism. Therefore, co-evolution mechanism should be introduced to the AIS; not only considering the relationships among individuals, but also the relationship between population and the environment and between population and population. That is a new target of AIS research.

Therefore, to solve the shortcomings of the fuzzy *c*-mean that is sensitive to the initial clusters and easy to fall into local optimization and the traditional evolutionary computation and immune clonal selection methods that lacks of considering the collaboration between populations and slow convergence, an immune clustering algorithm based on co-evolution is proposed in this paper. First, the clonal selection method is used to achieve the competition within population. This step consists of three operators: clone operator, mutation operator and clone selection operator. Through the clone operator, the proportion of the individuals which have a higher fitness value in the population goes up. The mutation operator is used to avoid the optimization process falling into a local optimum. The following is the clone selection operator. This operator selects the individuals with high fitness values to reconstruct each population. The internal evolution of each population is completed during this process (Du and Jiao 2002; Liu et al. 2003). Second, co-evolution operation among populations is conducted. This step includes four operators: better solution set neighborhood crossover operator, cooperative operator, annexation operator and division operator. As we all know, a better solution may exist in the nearby area of a good solution. And the better solution set neighborhood crossover operator is used to achieve the function of a local search to find out better solutions. Cooperation operator and annexation operator are used to complete the cooperation and competition among populations. These two operators realize the information exchange among populations and confirm with the process of biological evolution. After performing the annexation operator, the scale of one population expands because one population merges with another. To make the algorithm iterate, we perform division operation. In the division operator, we introduce a little mutation operator. As the iteration of the algorithm, the mutation becomes smaller. So taking the diversity into consideration, the convergence speed is ensured at the same time. Finally, the evolutionary results are compared with the global best individual results, with a strategy called Elitist Preservation, to find out the individual who has the highest fitness value, that is, the result of clustering. This strategy is used to maintain the diversity of solutions and to avoid the loss of solutions.

The rest of the paper is organized as follows. In the Sect. 2 the fuzzy *c*-means algorithm is introduced briefly. Section 3 presents the proposed immune clustering algorithm based on co-evolution. In Sect. 4, we analyze the influence of parameters on the result of the experiment in our algorithm and test and analyze the clustering results of four algorithms on UCI datasets and artificial datasets. At last, we make a conclusion of this paper.

2 Related work

In this part, the fuzzy *c*-means algorithm (FCM) is introduced. FCM algorithm is a soft clustering method based on fuzzy theory (Dunn 1973). It differs from *k*-means hard clustering algorithm which divides the data into groups, and calculates the cluster centers to minimize the objective function. And each point belongs to a group fixedly. However, the result of FCM algorithm is a membership degree matrix in which the values are between 0 and 1. And the values of membership degree matrix are used to determine the extent of one point belonging to various groups. According to the membership degree matrix, the respective data are placed into their corresponding cluster. This is similar to the natural environment and is very useful in calculating boundary point. Membership can be more accurate to describe the relationships between data and clusters, which enables us understand the deep inner relationship among data. The process of fuzzy *c*-means algorithm is as follows: at first initialize the cluster centers randomly; iterate the membership degree matrix U and cluster center matrix C to minimize the objective function value; complete the process of clustering.

FCM is based on the minimum value of the objective function of the following formula (1):

$$J_m = \sum_{j=1}^{k} \sum_{i=1}^{N} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \quad 1 < m < \infty \tag{1}$$

where m is a weight: a real number and greater than 1. This article takes $m = 2$. u_{ij} is the membership degree matrix, indicating the membership degree of x_i in cluster c_j. $\|x_i - c_j\|^2$ indicates the distance between data x_i and cluster center c_j. k is the number of clusters and N is the number of data.

The process of membership degree matrix u_{ij} iterates as following formula (2):

$$u_{ij} = \frac{1}{\sum_{p=1}^{k} \left(\frac{\|x_i - c_i\|}{\|x_i - c_p\|} \right)^{\frac{2}{m-1}}} \tag{2}$$

Cluster center matrix c_j iterates as following formula (3):

$$c_j = \frac{\sum_{i=1}^{N} u_{ij} \cdot x_i}{\sum_{i=1}^{N} u_{ij}} \tag{3}$$

The algorithm iterates the above two formulas to update the membership degree matrix U and cluster center matrix C, so as to minimize the objective function values. When $(\max\{|u_{ij}^{t+1} - u_{ij}^{t}|\}) < \varepsilon$, the iteration stops, where t is the number of iteration; ε is a real number and is greater than 0 and less than 1. When the iteration stops, the algorithm converges to a local minimum value of J_m.

When FCM algorithm converges, a group of cluster centers theoretically will be obtained and identify the classes to which all data belong through the membership degree matrix U. However, FCM is a local search method, and the final result depends closely on the initial selected points. If the initial points selected improperly, the algorithm is easy to find local optimal solution rather than the global optimal solution. But this problem can be just solved with evolutionary algorithm.

3 Co-evolution-based immune clonal algorithm for clustering

To solve the shortcomings of FCM and the traditional evolutionary computation, an algorithm called clustering algorithm based on co-evolution (ICCE) is put forward. It combines the advantages of FCM algorithm simplicity and fast convergence and the advantages of co-evolutionary algorithm, not depending on the initial population and able to find the global optimum.

3.1 Fitness function

To meet the requirements of evolutionary algorithm to select individuals with high fitness, the following formula (4) fitness function (Liu et al. 2012) is used:

$$\text{Fitness}(x_i) = \frac{c}{J_m + d} \tag{4}$$

Here, J_m is the objective function of FCM. c and d are constants and greater than zero. c is used to scale the objective function value, so that it is convenient to observation and $c = 100$ in this paper. d is used to guarantee the denominator is not 0.

3.2 Initialization

The populations X and Y are initialized as formulas (5) and (6) shown below:

$$X = \{C_{x1}; C_{x2}; \ldots\ldots; C_{xM}\} \tag{5}$$

$$Y = \{C_{y1}; C_{y2}; \ldots\ldots; C_{yN}\} \tag{6}$$

The size of population X is M and that of population Y is N. M and N can be different or be the same.

It is shown that the individual formats are initialized in formulas (5) and (6) below:

$$C_{xi} = \{x_i^{11}, x_i^{12}, \ldots, x_i^{1d}, x_i^{21}, x_i^{22}, \ldots, x_i^{2d}, \ldots\ldots, x_i^{K1}, x_i^{K2}, \ldots, x_i^{Kd}\}, \quad i \in [1, M] \tag{7}$$

$$C_{yj} = \{y_j^{11}, y_j^{12}, \ldots, y_j^{1d}, y_j^{21}, y_j^{22}, \ldots, y_j^{2d}, \ldots\ldots, y_j^{K1}, y_j^{K2}, \ldots, y_j^{Kd}\}, \quad j \in [1, N] \tag{8}$$

where K is the amount of clusters, d is the dimension of the data, and x_i and y_j are respectively the cluster centers of each dimension of the data. x_i and y_j are generated as random decimal numbers. The range of the initialization is between the maximum value and minimum value of each dimension in a data object.

3.3 Cloning and clonal selection operation

3.3.1 Clone operation

For the population X, considering the fitness value of the ith individual C_{xi} in the population, clone operation executes in accordance with the formula (9) rules (Du and Jiao 2002; Liu et al. 2003):

$$N_{\text{clone}}(i) = \text{int}\left(\mathrm{N_c} \cdot \frac{\text{fitness}(C_{xi})}{\sum_{j=1}^{N} \text{fitness}(C_{xj})} \right), \quad i = 1, 2, \ldots, M \tag{9}$$

where $N_{\text{clone}}(i)$ represents the amount of clone for the ith individual. N_c is a constant, which means the size of population after clone. Fitness(C_{xi}) represents the fitness of the ith individual in the population. int(y) represents the smallest integer which is greater than or equal to y. The determination method of clone amount in population Y is consistent with that of X.

After clone operation, the formats of populations X and Y are as shown in formulas (10) X_c and (11) Y_c below:

$$X_c = \{\underbrace{C_{x1}; C_{x1}; \ldots, C_{x1};}_{N_{\text{clone}}(1)} \underbrace{C_{x2}; C_{x2}; \ldots, C_{x2};}_{N_{\text{clone}}(2)} \underbrace{\ldots\ldots; C_{xM}; C_{xM}; \ldots, C_{xM}\}}_{N_{\text{clone}}(M)} \tag{10}$$

$$Y_c = \{\underbrace{C_{y1}; C_{y1}; \ldots, C_{y1};}_{N_{\text{clone}}(1)} \underbrace{C_{y2}; C_{y2}; \ldots, C_{y2};}_{N_{\text{clone}}(2)} \underbrace{\ldots\ldots; C_{yM}; C_{yM}; \ldots, C_{yM}\}}_{N_{\text{clone}}(N)} \tag{11}$$

As mentioned above, first calculate the individuals' fitness values and the needed clone amount of each individual in a population according to formula (9), and finally the individuals are cloned. After clone operation, the amount of individuals with high fitness values in the new population rises.

3.3.2 Mutation operation

Assuming that x_{ij} is a component of some individual in population X_c, x'_{ij} is a component of this individual after mutation operation. In accordance with the formula (12) (Liu et al. 2012) rules for mutation:

$$x'_{ij} = \begin{cases} 2 \cdot \alpha, & x_{ij} = 0, \alpha \geq p_m \\ -2 \cdot \alpha, & x_{ij} = 0, \alpha < p_m \\ (1 + 2 \cdot \alpha) \cdot x_{ij}, & x_{ij} \neq 0, \alpha \geq p_m \\ (1 - 2 \cdot \alpha) \cdot x_{ij}, & x_{ij} \neq 0, \alpha < p_m \end{cases} \tag{12}$$

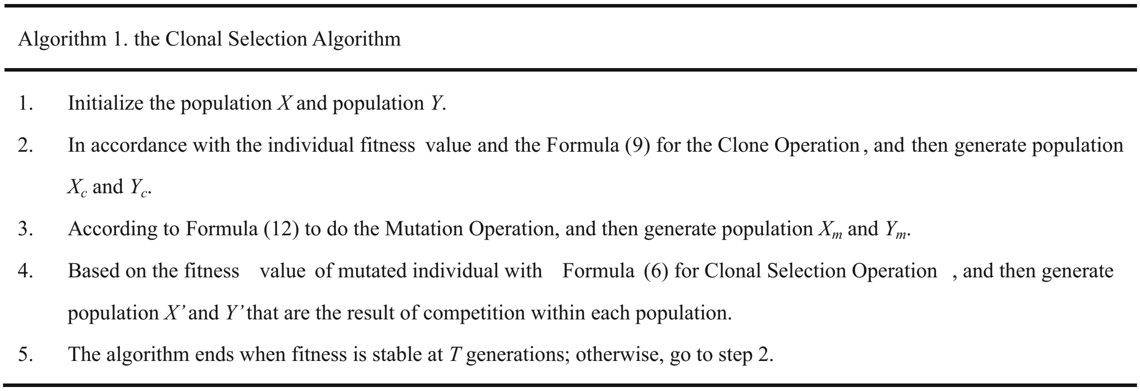
where α is a random constant between 0 and 1. p_m is the mutation probability and is also a constant. Its value is between 0 and 1. Here, the individuals before mutation operation are replaced with the individuals after that. The mutation method of population Y_c is consistent with that of X_c. The names of populations X_c and Y_c alter to X_m and Y_m then.

3.4 Clonal selection operation

For the new population X_m after clone operation and mutation operation, its scale is N_c. Suppose that $C_{xi}(t)$ is the individual before clone operation and mutation operation at tth generation and $C'_{xi}(t)$ is the individual after that. According to Formula (13) rules (Liu et al. 2012) to generate a selection probability value of an individual:

$$p_{CS}(C_{xi}(t+1)) = C'_{xi}(t)$$
$$= \begin{cases} 0, & \text{fitness}(C'_{xi}(t)) \leq \text{fitness}(C_{xi}(t)), C_{xi}(t) = \text{best}C_{xi}(t) \\ e^{\left(-\frac{\text{fitness}(C_{xi}) - \text{fitness}(C'_{xi})}{b}\right)}, & \text{fitness}(C'_{xi}(t)) \leq \text{fitness}(C_{xi}(t)), \text{and } C_{xi}(t) \neq \text{best}C_{xi}(t) \\ 1, & \text{fitness}(C'_{xi}(t)) > \text{fitness}(C_{xi}(t)) \end{cases} \tag{13}$$

where b is a constant that is greater than zero. p_{cs} is the probability that C'_{xi} replaces C_{xi}. According to the following rules for clonal selection operation: if $p_{CS} = 0$, C_{xi} still remains itself after clonal selection operation; if $p_{cs} = 1$, C_{xi} is replaced by C'_{xi} after clonal selection operation; If $p_{cs} \neq 0$ and $p_{cs} \neq 1$, generate a random number between 0 and 1 and compare r and p_{cs}. When $r \geq p_{cs}$, operate as that of $p_{CS} = 0$, and when $r < p_{cs,}$ operate as that of $p_{CS} = 1$. Clonal selection operation for population Y_c is consistent with that for X_c. The scale of the population returns to its previous state after Clonal selection operation. After this operation, we suppose the populations are X' and Y'. Algorithm 1 shows the clonal selection process.



Algorithm 1. the Clonal Selection Algorithm

```
1. Initialize the population $X$ and population $Y$.
2. In accordance with the individual fitness value and the Formula (9) for the Clone Operation, and then generate population $X_c$ and $Y_c$.
3. According to Formula (12) to do the Mutation Operation, and then generate population $X_m$ and $Y_m$.
4. Based on the fitness value of mutated individual with Formula (6) for Clonal Selection Operation , and then generate population $X'$ and $Y'$ that are the result of competition within each population.
5. The algorithm ends when fitness is stable at $T$ generations; otherwise, go to step 2.
```

3.5 Better solution set neighborhood crossover operator

For the current population X', its individuals are C_{xi}, $i = 1, 2, \ldots, M$. The former percentages $a(a_1, a_2)$ of best individuals in the population constitute the set P, which is called a better solution set. And its individual is p_{xj}, $j = 1, 2, \ldots, a \cdot M$. Randomly select an individual p_{xj}, $j \in [1, a \cdot M]$ from P, and according to the following formula (14) (Jiao et al. 2012) to generate a new solution $C_{xip}$to replace the original solution C_{xi}:

$$C_{xip} = p_{xj} + G(0, 1) \cdot (p_{xj} - C_{xi}), \quad i \in [1, M] \tag{14}$$

where $G(0, 1)$ represents a random number generator of Gaussian distribution. $a(a_1, a_2)$ is a constant, which is greater than 0 and less than 1.

As we all know, there may be a better solution near a good solution. In fact, the better solution set neighborhood crossover operator is used to search for these solutions in a local searching way.

In this algorithm, the total execution time of the better solution set neighborhood crossover operator is $p_{c1} * M$, and p_{c1} is a real number which is greater than 0 and less than 1. The better solution set neighborhood crossover operator for population Y_c is consistent with that for X_c. After this operation, the populations X' and Y' alter to X'' and Y''.

3.6 Cooperation operator

After better solution set neighborhood crossover operator, cooperation operation or annexation operator is executed. The main role of this operator is to introduce the better individuals of one population into another. Thus, the update rate of population is accelerated, and the phase of falling into local optimum can be avoided. For the two populations X'' and Y'', their individuals are C''_{xi} and C''_{yj}, respectively. The former percentages a_1 and a_2 of better individuals in the populations X'' and Y'' constitute the set P_x and P_y, which are called better solution sets. And their individuals are p_{xk} ($k = 1, 2, \ldots, a_1 * M$) and p_{yl} ($l = 1, 2, \ldots, a_2 * N$). Randomly select two individuals $p_{xk}, k \in [1, a_1 * M]$ and p_{yl}, $l \in [1, a_2 * N]$ from P_x and P_y, and then according to the following formulas (15) and (16) to cooperate the two populations (Jiao et al. 2012):

$$C'''_{xi} = p_{yl} + G(0, 1) \cdot (p_{yl} - C''_{xi}), \quad i \in [1, M] \tag{15}$$

$$C'''_{yj} = p_{xk} + G(0, 1) \cdot (p_{xk} - C''_{yj}), \quad j \in [1, N] \tag{16}$$

wherein C'''_{xi} and C'''_{yi} represent the individuals in population X''' and population Y''' after cooperation operation. M and N are the sizes of two populations. $G(0, 1)$ represents a random number generator of Gaussian distribution. The new individuals replace the old ones. And here, the total execution time of the cooperation operator is $p_{c2} * n$, and n is the size of the population. p_{c2} is a real number which is greater than 0 and less than 1.

X and Y are two evolved populations separately and search for solutions only in their own populations. Here, p_{yl} is a better individual; through the cooperation operator (15), information of population Y is exchanged to population X and the search space of the algorithm is expanded while still maintaining the diversity of population (so does p_{xk}). After the Cooperation Operator, the populations X''' and Y''' are written as X_{new} and Y_{new}.

3.7 Annexation operator

If one population is superior to the other one, it is not necessary for all these two populations to evolve. Then execute the annexation operator, and the superiority population merges with the inferiority population and then produces a new population X''' or Y'''. The individuals of them are C'''_x and C'''_y. Here, we suppose that the population X'' is superior to Y''. The principle we judge which population is superior is that the former percentage b_1 of fitness values in population X'' are greater than that b_2 in population Y''. Randomly select two individuals from P_x and Y'', called $p_{xi}, i \in [1, a_1 * M]$ and C''_{yj}, $j \in [1, N]$. And then operate according to the following formulas (17) and (18) (Jiao et al. 2012) for the inferiority population:

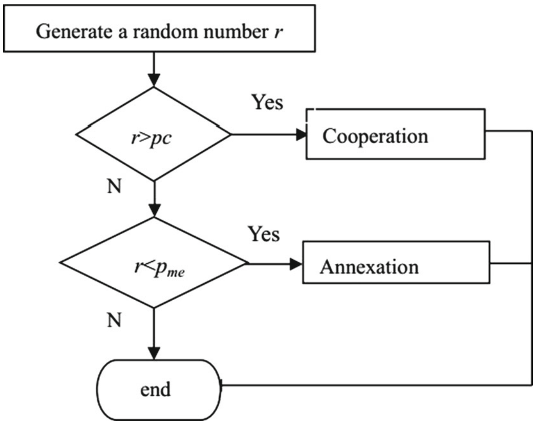


Fig. 1 Judgment process of execute cooperation operator or annexation operator

$$C'''_{yj} = p_{xi} + G(0,1)\cdot(p_{xi} - C''_{yk}), \quad i \in [1, a_1 * N], \quad j \in [1, N] \tag{17}$$

The formula (17) is the operation that X'' merges with Y''. In turn, the operation that Y'' merges with X'' is as follows:

$$C'''_{xi} = p_{yj} + G(0,1)\cdot(p_{yj} - C''_{xi}), \quad i \in [1, M], \quad j \in [1, a_2 \cdot N] \tag{18}$$

After this operation, the superiority population and the fresh inferiority population are combined to form a new population called Z, and its size is $M + N$.

Here, the total execution time of the annexation operator is $p_{c3} * n$, and n is the size of the population. Namely execute $p_{c3} * M$ times for population X'' and $p_{c3} * N$ times for Y''.

Although the inferiority population is less likely to obtain the optimal solution, it still has some useful information. Through the above formula (17) and (18), we keep this information. Whether the cooperation operator or annexation operator carried out is in accordance with Fig. 1.

3.8 Division operator

After the annexation operator, the size of the population alters to $M+N$. The division operator is used to ensure the iteration of the algorithm. After that, the new population Z is split into two sub-populations. Here, the new population Z is randomly divided into two sub-populations X_{new} and Y_{new} with the size M and N, and perform the following operation to one sub-population. Assuming the population is X_{new} and its scale is M and the individuals in the population are represented as C_{xi}, $i = 1, 2, \ldots, M$. According to the formula (19):

$$C_{xi} = \begin{cases} C_{xi}, G(0,1) < \frac{1}{n} \\ C_{xi} + U(0, \frac{1}{t}), G(0,1) \geq \frac{1}{n} \end{cases}, \quad i = 1, 2, \ldots, M \tag{19}$$

Here, $G(0,1)$ and $U(0, \frac{1}{t})$ represent random number generators of Gaussian and uniform distribution, respectively. t is the time of iteration. The above operation (19) is equivalent to a small mutation, and thus the diversity of population is maintained so that the premature convergence of the algorithm is avoided. The operation of population Y_{new} is consistent with that of X_{new}.

3.9 Elitist preservation strategy

After all co-evolution operations, the percentage p_x of outstanding individuals are reserved to form the global outstanding individual group, namely the elite group. It is used to compare with the outstanding individuals in next generation to select the amount of $px \cdot (M + N)$ individuals to form the new elite group. Then, the iteration goes on. This strategy executes once in each of the next iterations then. It is helpful to maintain the solution diversity, to avoid loss of solution and to find out the global optimal solution.

And here is a clearly explanation about the concept of co-evolution. The cooperation operator, the annexation operator and division operator are different from the concepts in clonal selection and affinity maturation. In clonal selection and affinity maturation, the operators are operated among individuals. However, in co-evolution, the concepts are used among populations. They are the operators of populations, not of the individuals. These operators are used to promote the mutual evolution of populations. In fact, in the environment, the collaboration is a common phenomenon. Those operators are the collaborative operators of co-evolution algorithm.

3.10 The overall process of the proposed algorithm

Algorithm 2 gives the overall process of the proposed algorithm in the paper.

Algorithm 2. A clustering algorithm based on immune co-evolution

1. Initialize two cluster center populations X and Y, and set the parameters.
2. For the cluster center populations, Cloned, Mutation and Clonal Selection Operation are executed. Then produces optimized populations X' and Y' .
3. Optimizing population X' and Y' according to co-evolution operations. Firstly execute the Better Solution Set Neighborhood Crossover Operator to generate the population X'' and Y''; secondly judge to execute the Cooperative Operator or Annexation Operator and Division Operator according to Figure 1 to generate new populations X_{new} and Y_{new}.
4. Execute the Elite Preservation Strategy. Compare the fitness values of the best individual group in old population and that in the new population; update the global best individual group and record the best individual and its fitness value; replace the old population with the new one.
5. If the termination condition of iteration is satisfied, the algorithm terminates. Otherwise, go to step 2. The termination condition may be a certain number of iteration, such as T_1, and it also can be the fitness value of the best individual does not change in T_2 generations. T_1=200 is chosen in this paper's experiments.
6. According to the U matrix to judge the final clusters which all the data belong to

Moreover, in the implementation process of the algorithm, if the cluster center matrix changes, FCM clustering algorithm is executed iteration once to speed up the convergence.

3.11 Complexity analysis

The computational complexity of all operators is analyzed in this part. Here, N is the amount of the data and d is the dimension of the data. K is the number of real clusters, n is the amount of the individuals in a population. Nc, p_{c1}, p_{c2} and p_{c3} are as that of 3.3.1, 3.5, 3.6 and 3.7 defined.

The complexities of the main operators are shown as follows. The computational complexity of Initialization operator is $O(K * d * n)$. The computational complexity of clone and the division operators is $O(n)$. The computational complexity of mutation and clonal selection operators is $O(d * K * Nc)$. The computational complexities of better solution set neighborhood crossover, cooperation and annexation operators are $O(d * K * p_{c1} * n)$, $O(d * K * p_{c2} * n)$ and $O(d * K * p_{c3} * n)$, respectively. The computational complexity of fitness operator is $O(kdnN)$. Thus, the worst computational complexity of the proposed algorithm is $O(K * d * n) + 2O(n) + O(d * K * Nc) + O(d * K * p_{c1} * n) + O(d * K * p_{c2} * n) + O(d * K * p_{c3} * n) + O(k * d * n * N) = O(K * d * n + n + d * K * Nc + d * K * p_{c1} * n + d * K * p_{c2} * n + d * K * p_{c3} * n + k * d * n * N)$.

Because of $N \gg n$, k, d, and according to the analysis above, the total computational complexity can be simplified as $O(k * d * n * N)$. Therefore, reducing the time of the fitness function should be considered when dealing with big data with the proposed algorithm.

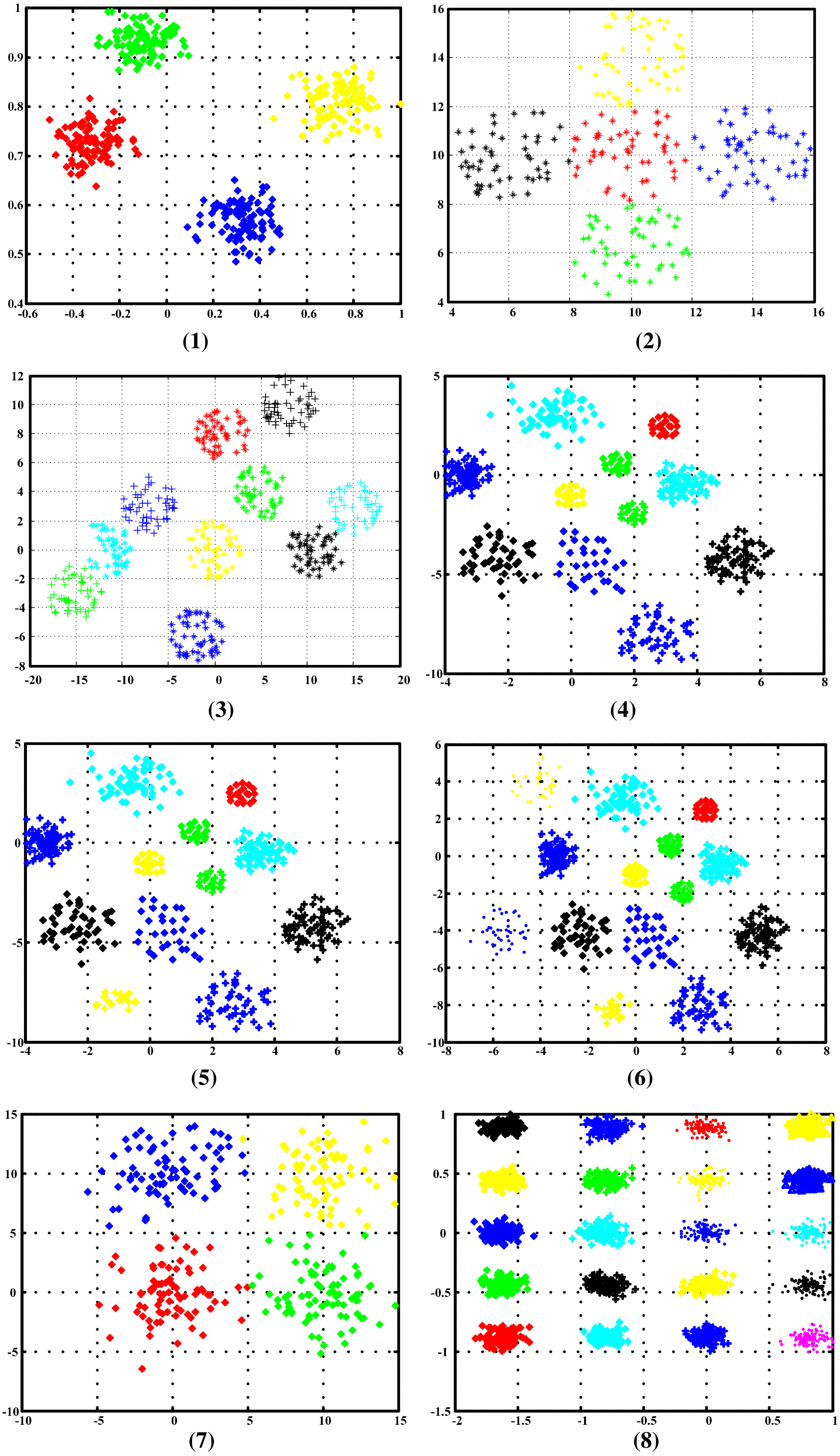
4 Experiments and analyses

Three algorithms are implemented in Matlab R2010a on HP dc7800 (Intel(R) Core(TM) 2 Duo CPU and the system of Microsoft Windows 7). That are Inmmunodomaince-based Clonal Selection Clustering Algorithm (ICSCA) (Liu et al. 2012), Inmmunodomaince-based Clonal Selection Clustering Algorithm with Elitist preservation (ICSCAE) and the proposed algorithm (ICCE).

4.1 Datasets

Each algorithm is tested on eight UCI datasets and eight artificial datasets. The UCI datasets are *iris*, *wine*, *seeds*, *lung_cancer*, *vote*, *sonar,* a sample script dataset—Handwritten Digital and *Spambase* (http://archive.ics.uci.edu/ml/datasets.html). Figure 2 shows the artificial datasets.

It can be seen from Fig. 2 that data01 is a basic problem with four clusters. Its boundary is clear and easy to distinguish. data02 is a blurred boundaries clustering problem. data03, data04, data05, and data06 are clustering problems which have a number of different sizes and different distributions of clusters. data07 is a clustering problem with discrete points. data08 is a clustering problem with 20 clusters and 2,000 data points, which is a multi-point and multi-class problem.



(1) **(2)**

(3) **(4)**

(5) **(6)**

(7) **(8)**

Fig. 2 Artificial datasets. **1** data01, **2** data02, **3** data03, **4** data04, **5** data05, **6** data06, **7** data07, **8** data08

For example, there are three classes in the datasets called 1, 2, 3 and A, B, C are the data sets of each class, that is, A is the data set of first class with real labels 1, B is the second with 2 and C is the third with 3. After the clustering, there may be a result as that A is the third cluster, B is the first and C is the second. According to this result, each of A, B and C has an incorrect cluster. Actually, we have clustered A, B and C correctly. So, the mapping relation between the labels obtained by clustering and the real labels should be defined.

Clustering is an unsupervised process without a training process. In this manuscript all the test datasets have a real class value, and the real values are used to calculate the accuracy in the experiment.

The following approach is used to define the mapping relation: to select the majority of the real label in a cluster as the new label of the cluster. Thus, A is the first class, B is the second, and C is the third.

In fact, the accuracy is defined as most literatures do. It is the ratio of the size of the correct clustering units and the size of the data. The 'mean accuracy' is computed with the mean accuracies of 30 independent runs.

Suppose N_1 is the amount that the new labels are same with the actual label of a data, and the size of dataset is N. The accuracy is calculated as formula (20):

$$\text{accuracy} = \frac{N_1}{N} \tag{20}$$

Then the way to calculate the mean accuracy meanacc is shown in formula (21):

$$\text{meanacc} = \frac{1}{m}\sum_{i=1}^{m}\text{accuracy}(i) \tag{21}$$

where m is the times of experiments; accuracy(i) is the cluster accuracy in the ith experiment. Similarly, labels are added to each datum in the generation of artificial two-dimensional datasets.

Script dataset comes from the UCI dataset "Optical Recognition of Handwritten Digital Dataset" (http://archive.ics.uci.edu/ml/datasets.html). 120 records are randomly selected to constitute the Handwritten Digit dataset, called "digit" simply. The numbers of the digit dataset are 2, 5, 6 and 9. Each number selects 30 records. A part of the digit dataset is shown in Fig. 3.

4.2 Parameter analysis

The parameters used in this paper are as follows: M and N for the sizes of populations X and Y, respectively; the scale N_c of clone operation; the mutation probability P_m; the parameter b in clonal select operation; the percentages a_1 and a_2 of the better solution group; the percentage p_{c1} in the better solution set neighborhood crossover operator; the probability p_{co} and p_{merge} of cooperative operator and annexation operator; the percentage p_{c2} in cooperative operator; the percentage p_{c3} in annexation operator; the percentage p_x in Elitist Preservation Strategy.

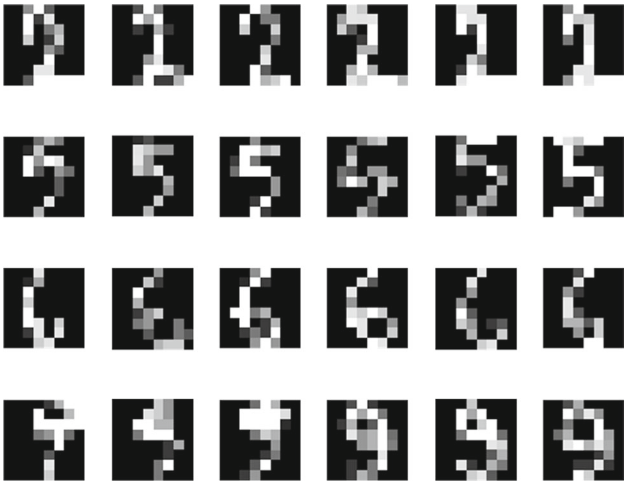


Fig. 3 Handwritten dataset "digit"

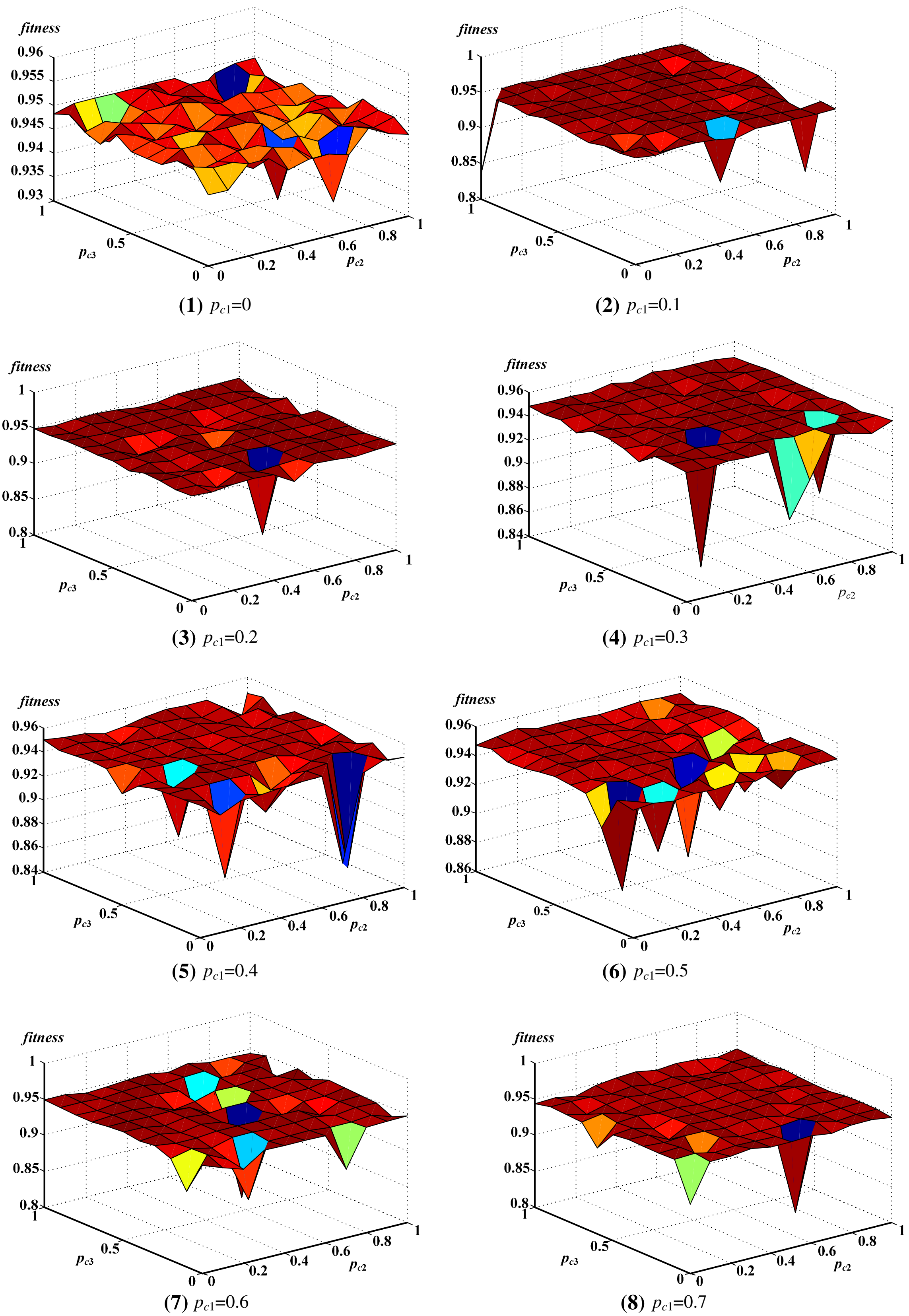
In the experiments of this paper, the parameter values of clonal selection are selected as per the reference (Liu et al. 2012). Namely, $M = 50$, $N = 50$, $N_c = 80$, $p_m = 0.3$ and $b = 0.16$. On the basis of the experience, the percentages a_1, a_2 and p_x are determined in [0.1, 0.2]. Here, choose $a_1 = a_2 = 0.1$, $p_x = 0.1$. And the probability $p_{co} = 0.7$ and $p_{\text{merge}} = 0.3$. The parameters p_{c1}, p_{c2} and p_{c3} are analysis below. p_{c1} is selected from 0.1 to 0.7 with interval 0.1. So the total amount of p_{c1} is eight. For each p_{c1}, p_{c2} and p_{c3} are selected from 0 to 1, with the interval 0.1. So both p_{c2} and p_{c3} have 11 values. Thus for each p_{c1} value, there are 121 corresponding fitness values. Here, the proposed algorithm ICCE is performed on dataset wine. And p_{c2} and p_{c3} are x and y coordinates; the fitness value fitness is z coordinate. The averaged fitness value of ten times is taken as fitness. In the analysis process, the other parameters are as mentioned above. Three-dimensional graphs are shown in Fig. 4.

It can be seen from Fig. 4 that the result of the algorithm is relatively stable when p_{c1} is 0.1 and 0.2. In this case, the maximum fitness value is at $p_{c1} = 0.1$, $p_{c2} = 0.3$, $p_{c3} = 0.3$ and $p_{c1} = 0.1$, $p_{c2} = 0.2$, $p_{c3} = 0.7$ and the average fitness value fitness = 0.9515. Noting that the larger p_{c1}, p_{c2}, p_{c3} are, the greater calculation amount is, then the parameters $p_{c1} = 0.1$, $p_{c2} = 0.3$, $p_{c3} = 0.3$ are chosen.

4.3 Experimental results and analysis

4.3.1 The analysis of main operators

The references (Al-Muallim and El-Kouatly 2010; Liu et al. 2012) have proven the effectiveness of some operators. However, there are also some limitations in these operators. Thus, to improve the effective, three co-evolution operators are pro-



(1) p_{c1}=0

(2) p_{c1}=0.1

(3) p_{c1}=0.2

(4) p_{c1}=0.3

(5) p_{c1}=0.4

(6) p_{c1}=0.5

(7) p_{c1}=0.6

(8) p_{c1}=0.7

Fig. 4 The relationship between the average fitness value and p_{c1}, p_{c2}, p_{c3}

Table 1 The mean accuracy of 30 independent runs of the proposed algorithm without some operator

Dataset	A	B	C
Iris	0.9558 (0.0092)	0.9567 (0.0065)	**0.9582 (0.0043)**
Wine	0.9481 (0.0028)	**0.9487 (0.0019)**	0.9466 (0.0051)
Seeds	0.8929 (0.0058)	0.8929 (0.0063)	**0.8935 (0.0058)**
Lung_cancer	**0.5067 (0.0581)**	0.4917 (0.0525)	0.5063 (0.0515)
Vote	**0.8620 (0.0004)**	0.8602 (0.0034)	0.8606 (0.0035)
Sonar	**0.5545 (0.0144)**	0.5529 (0.0204)	0.5494 (0.0139)
Digit	**0.7794 (0.0553)**	0.7617 (0.0687)	0.7586 (0.0510)
Spambase	0.7339 (0.0463)	**0.7351 (0.0290)**	0.7324 (0.0570)
data01	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
data02	0.9437 (0.0280)	0.9535 (0.0084)	**0.9535 (0.0052)**
data03	0.9977 (0.0016)	0.9978 (0.0006)	**0.9981 (0.0008)**
data04	**0.9768 (0.0275)**	0.9702 (0.0205)	0.9746 (0.0177)
data05	0.9885 (0.0231)	0.9888 (0.0219)	**0.9905 (0.0203)**
data06	0.9700 (0.0512)	0.9815 (0.0352)	**0.9855 (0.0337)**
data07	0.9839 (0.0142)	0.9854 (0.0049)	**0.9864 (0.0015)**
data08	0.9947 (0.0216)	0.9896 (0.0571)	**0.9956 (0.0145)**

Bold values represent best result

A: The proposed algorithm without Better solution set neighborhood crossover operator

B: The proposed algorithm without cooperative operator

C: The proposed algorithm without annexation operator and division operator

posed in the proposed algorithm. They are better solution set neighborhood crossover operator, cooperative operator, and annexation operator and division operator. The effectiveness of the three operators is evaluated in this section. Table 1 gives the mean accuracy of 30 independent runs of the proposed algorithm without one of these three operators to test the contribution of each operator to the proposed algorithm.

It can be seen from Table 1 that the proposed algorithm without better solution set neighborhood crossover operator has the highest mean accuracies in five of the datasets, the proposed algorithm without cooperative operator has the highest mean accuracies in two datasets, and the proposed algorithm without annexation operator and division operator has the highest mean accuracies in eight datasets. That means the Annexation Operator and Division Operator are the most important operators in the algorithm, followed by better solution set neighborhood crossover operator, and finally, cooperative operator.

Table 2 The comparative test results of the highest accuracy fitness in various algorithms

Dataset	Amount of dataset	ICSCA	ICSCAE	ICCE
Iris	150	0.9600	0.9600	**0.9667**
Wine	178	0.9551	0.9551	**0.9663**
Seeds	210	0.9048	0.9000	**0.9286**
Lung_cancer	32	0.5938	0.5938	**0.6250**
Vote	435	0.8713	0.8713	**0.8805**
Sonar	208	0.6058	0.6298	**0.6875**
Digit	120	0.8500	0.8917	**0.9500**
Spambase	4,601	0.7448	0.7659	**0.8059**
data01	400	1.0000	1.0000	1.0000
data02	300	0.9600	0.9680	**0.9720**
data03	250	1.0000	1.0000	1.0000
data04	515	0.9961	0.9961	**0.9981**
data05	535	0.9981	0.9963	**1.0000**
data06	605	0.9950	0.9950	**0.9983**
data07	320	0.9875	0.9906	**0.9969**
data08	2,000	1.0000	1.0000	1.0000

Bold values represent best result

Table 3 The average accuracies of the results of 30 independent runs and the corresponding standard deviations

Dataset	ICSCA	ICSCAE	ICCE
Iris	0.9564 (0.0052)	0.9544 (0.0058)	**0.9589** (0.0031)
Wine	0.9431 (0.0072)	0.9419 (0.0052)	**0.9493** (0.0018)
Seeds	0.8938 (0.0036)	0.8930 (0.0039)	**0.8962** (0.0064)
Lung_cancer	0.5021 (0.0666)	0.4990 (0.0430)	**0.5115** (0.0529)
Vote	0.8608 (0.0040)	0.8608 (0.0036)	**0.8634** (0.0045)
Sonar	0.5438 (0.0121)	0.5479 (0.0226)	**0.5622** (0.0261)
Digit	0.7178 (0.0605)	0.7044 (0.0647)	**0.8310** (0.0627)
Spambase	0.6444 (0.0509)	0.6610 (0.0518)	**0.7366** (0.0628)
data01	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
data02	0.9500 (0.0040)	0.9507 (0.0042)	**0.9557** (0.0088)
data03	0.9985 (0.0009)	**0.9987** (0.0010)	0.9981 (0.0006)
data04	0.9771 (0.0181)	0.9757 (0.0180)	**0.9814** (0.0201)
data05	0.9814 (0.0172)	0.9790 (0.0175)	**0.9915** (0.0112)
data06	0.9846 (0.0150)	**0.9899** (0.0118)	0.9874 (0.0379)
data07	0.9875 (0.0000)	0.9875 (0.0000)	**0.9879** (0.0037)
data08	0.9950 (0.0201)	0.9850 (0.0267)	**0.9980** (0.0092)

Bold values represent best result

4.3.2 Experimental results and analysis

The evaluation indexes are the highest accuracy fitness, the average accuracy meanacc of 30 independent runs and the standard deviation std, and the *t* test results. Table 2 shows the highest accuracy fitness of various algorithms. Table 3 shows the meanacc (std) of algorithms. Table 4 shows the *t* values between the proposed algorithm ICCE and the compared algorithms with a significance level of 0.05.

It can be seen that the highest accuracies of the proposed algorithm are higher than other two algorithms in eight UCI datasets from Table 2. For datasets *wine*, *seeds*, *lung_cancer*, *sonar, digit* and *spambase*, the highest accuracy of the pro-

Table 4 The t values between ICCE and the compared algorithms

Dataset	Iris	Wine	Seeds	Lung_cancer	Vote	Sonar	Digit	Spambase
ICCE/ICSCA	1.9867[a]	4.5687[a]	1.0648[b]	0.6037[b]	2.2895[a]	4.3842[a]	2.9902[a]	6.2508[a]
ICCE/ICSCAE	3.3559[a]	7.3048[a]	1.5683[b]	1.0046[b]	2.4866[a]	2.5598[a]	3.6461[a]	5.0886[a]
Dataset	data01	data02	data03	data04	data05	data06	data07	data08
ICCE/ICSCA	–	3.2296[a]	0.0000[b]	0.8784[b]	1.7972[a]	−0.0267[b]	0.6107[b]	0.9074[b]
ICCE/ICSCAE	–	2.8280[a]	−0.8234[b]	1.1709[b]	2.2966[a]	−1.0853[b]	0.6107[b]	3.0460[a]

Bold values represent best result

[a] ICCE is significantly better than the compared algorithm

[b] ICCE has no significant performance difference with the compared algorithm

posed algorithm is 0.01–0.06 higher than the other three algorithms. Especially for datasets *lung_cancer digit* and *spambase*, the accuracy is 0.04–0.06 higher than all the other algorithms. On dataset digit, the accuracy of the proposed algorithm is 0.6 higher than other two algorithms. This indicates that a better result is achieved by the proposed algorithm in high-dimensional and more complex dataset than the other two algorithms. In the artificial datasets, the proposed algorithm can accurately separate the clusters to which all the points belong. For dataset data2, which boundary is not clear, the accuracy got by the proposed ICCE is the highest in the four algorithms, showing that ICCE has the ability to deal with the dataset which has an unclear boundary. ICCE can find higher fitness value and will not fall into local optimum. For datasets data6 and data7, which are multi-clusters and with different sizes and distributions, the proposed algorithm clusters precisely show that ICCE has the ability to find small clusters and adapt to different distributions. The above analysis shows that the proposed algorithm has a better optimization searching capability.

It can be seen from Table 3 that in all the datasets, the average accuracies got by ICSCAE and ICCE are higher than that by ICSCA because these two algorithms use an Elitist Preservation Strategy. This experiment result proves that the Elitist Preservation Strategy is helpful to obtain a stability result of the algorithm. Comparing the average accuracies of ICSCAE and ICCE in these 16 datasets, it can be found that the average accuracies of ICCE are higher than that of ICSCAE in 14 datasets. It is because ICCE has the idea of co-evolution and these experiment results prove that the co-evolutionary algorithm stabilizes the whole algorithm. In particular dataset digit, average accuracy got by the proposed algorithm ICCE is 0.05 more than the two contrasting algorithms, which shows that ICCE algorithm is very effective for high-dimensional dataset. In artificial datasets, the average accuracies got by ICSCA are lower than ICSCAE and ICCE. This indicates that the algorithms with Elitist Preservation Strategy have a stable result in both UCI and artificial datasets. In these artificial datasets, the average accuracies got by ICCE are higher than ICSCAE in six datasets, which shows that the joint co-evolutionary algorithm is helpful in testing the stability of the clustering result for the artificial datasets. Especially for data8, which has 20 clusters and 2,000 data, is used in the experiment to test the validity of the algorithms for multi-clusters and large-scale dataset, the proposed ICCE has an average accuracy 1. It reveals that ICCE is effective with this kind of dataset.

To compare the clustering effect of ICCE and ICSCA and ICSCAE, t test is used here and the significance level $\alpha = 0.05$ is selected. First, all the three algorithms are run for 30 times randomly. Second, assume that the mean accuracy of ICCE is better than of ICSCA and ICSCAE. And the rejection region is $t \geq t_{0.05}(30 + 30 - 2) = 1.67065$. It means that ICCE is better than the corresponding compared algorithms: while t value in this region has no significant performance difference with the compared algorithm while out of the region. Table 4 gives the t values between ICCE and the compared algorithms.

It can be seen from Table 4 that the proposed ICCE algorithm is significantly better than the other algorithms in eight UCI datasets and two artificial datasets. For dataset *spambase*, whose size is 4,601, it is a large-scale dataset. It can be seen that the t values are large from Table 4. It indicates that the proposed algorithm ICCE has the ability to deal with a large-scale dataset. The t values are also large for datasets data02, which is a blurred boundaries clustering problem, which indicates that ICCE has a better ability to deal with the blurred boundaries clustering problems.

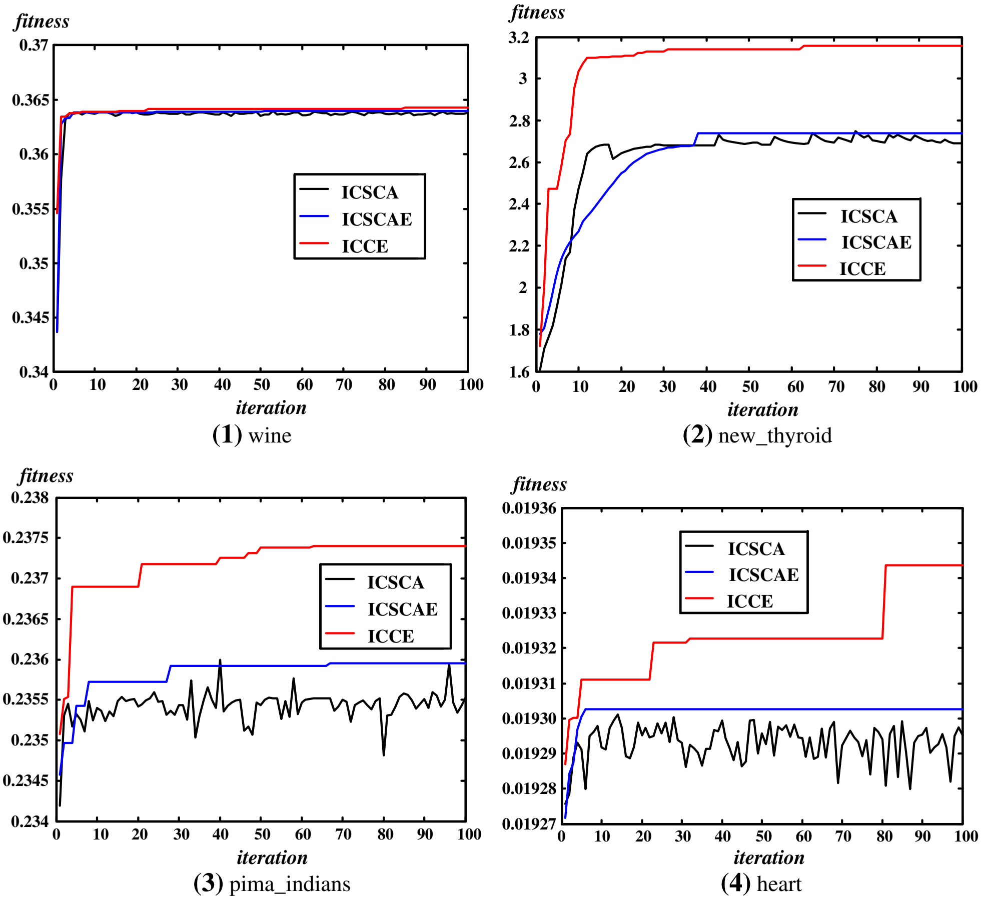
The reference (Liu et al. 2012) has been compared the effectiveness of ICSCA and FCM. So here some datasets are used to compare k-means, FCM, ICSCAE and the proposed algorithm.

It can be seen from Table 5 that the immune clonal algorithms have a better effect than k-means and FCM. Although most of the experimental results with the proposed algorithm are better than the other algorithms, the limitation of the proposed is that it could not deal with manifold data.

Table 5 The average accuracies of 30 independent runs and the corresponding standard deviations of FCM, ICSCAE and ICCE

Dataset	*k*-Means	FCM	ICSCAE	ICCE
Iris	0.8131 (0.2115)	0.8933 (0.0000)	0.9544 (0.0058)	**0.9589** (0.0031)
Vote	0.8579 (0.0478)	0.8621 (0.0001)	0.8608 (0.0036)	**0.8634** (0.0045)
Sonar	0.5442 (0.0121)	0.5530 (0.0015)	0.5479 (0.0226)	**0.5622** (0.0261)
data01	0.8455 (0.1799)	**1.0000 (0.0000)**	**1.0000 (0.0000)**	**1.0000 (0.0000)**
data03	0.8229 (0.1019)	0.9790 (0.0493)	**0.9987** (0.0010)	0.9981 (0.0006)
data05	0.7949 (0.0682)	0.8598 (0.0615)	0.9790 (0.0175)	**0.9915** (0.0112)
data07	0.9770 (0.0576)	0.9875 (0.0000)	0.9875 (0.0000)	**0.9879** (0.0037)

Bold values represent best result



(1) wine

(2) new_thyroid

(3) pima_indians

(4) heart

Fig. 5 The fitness value graph with the number of iteration in three algorithms

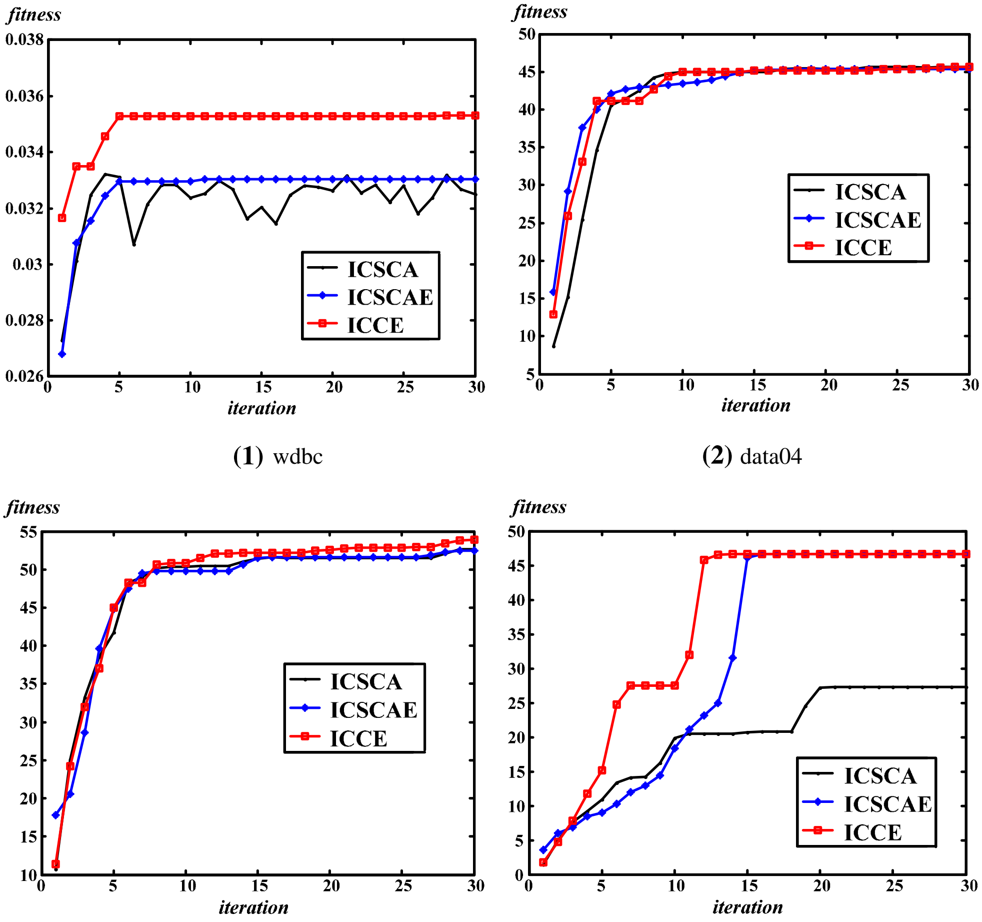
4.4 Convergence and diversity analysis

4.4.1 Convergence analysis

In this part, the fitness value curves of the three algorithms are given to observe the convergence of the algorithms in four UCI datasets (http://archive.ics.uci.edu/ml/datasets.html).

It can be seen from Fig. 5 that the fitness value curves of ICSCAE are fluctuated and in a state of non-convergence, but that of the ICSCAE and the proposed ICCE are stable and unchanged after several iterations. This indicates the role of the Elitist Preservation Strategy in maintaining the diversity of the solutions while retaining the excellent solutions. Especially for datasets *new_thyroid*, *pima_indians* and *heart*, it can be seen that the proposed ICCE has a largest fitness value than the other algorithms and the curve has a gradually upward trend, which show that the co-evolutionary algorithm plays a better role in maintaining the population diversity in the searching process for the global optimum solutions.

In addition, observing Fig. 5 carefully, it can be found that the number of iteration with ICCE is smaller than that with the other two algorithms while achieving the same fitness values. For example, for dataset *new_thyroid*, to achieve the fitness value 2.7, the proposed algorithm ICCE only needs 7



(1) wdbc **(2)** data04

(3) data06 **(4)** data08

Fig. 6 The comparison of diversity

times iteration, while ICACS needs 15 times and ICSCAE needs 35 times; for dataset *pima_indians*, to achieve the fitness value 0.236, the proposed algorithm ICCE only needs 3 times iteration, while ICACS needs 68 times and ICSCA does not converge. This shows that the ICCE algorithm converges faster than the other two algorithms. What is more, for dataset heart, it can be found that the ICSCAE algorithm falls into a local optimum situation but ICCE does not. This in another way shows that the ICCE algorithm is able to search for the global optimal solutions. All the above show that the proposed algorithm ICCE only needs a few iterations while getting a better result. These prove that the proposed algorithm can converge fast while achieving a higher fitness value.

4.4.2 Diversity analysis

In Fig. 6, ICSCA, ICSCAE and the proposed ICCE are compared to analysis the diversity of the algorithms in four datasets.

In Fig. 6, for dataset wdbc, which is special for the high-dimensional and complex data structures, the fitness value curve of ICSCAE is unstable while that of ICCE is stable step by step. The phenomenon can also be observed in datasets data4 and data6. Furthermore, in the iteration process, the fitness value of the proposed algorithm ICCE increases rapidly at the beginning, which shows the advantages of the Elitist Preservation Strategy, and when it is large this value increases slowly but do not stop, which shows the advantages of co-evolution operations in avoiding falling into a local optimum. These are the lack of the other two algorithms. For dataset data8, which size is 2,000 and has 20 different clusters, it can be noticed that ICCE converges faster than the other algorithms, which shows the advantages of ICCE in processing the large-scale dataset. Summarize the results of the algorithms on these four datasets, it can be seen that the proposed algorithm has the highest fitness values and the fastest convergence speed. This indicates that the proposed algorithm ICCE has the ability to find out the global optimal solutions, and proves that ICCE has a better diversity indirectly.

5 Conclusion

Clustering is an important mean in data mining, and FCM algorithm is one of the most classical clustering algorithms. It is based on fuzzy theory and has a good effect. What is more, it is easy to understand and implement. But the FCM algorithm is sensitive to the initialization of cluster centers and different initializations will lead to different clustering results. In addition, this algorithm is easy to converge to a local optimal solution. These problems can be solved with the evolutionary algorithms. But the traditional evolutionary algorithms simply emphasize the competition within populations while biological evolution in nature is not just an internal competition but what more important is the relationships between populations. Co-evolution algorithm is a newest branch of the evolutionary algorithm and is different from the traditional evolutionary algorithm which just underlines the struggle within the population. The co-evolution algorithm is a combination of cooperation and competition among populations and is more similar to the relationships among species in nature. To solve the existing problems of FCM and traditional evolutionary algorithms, we propose an immune clustering algorithm based on co-evolution (ICCE) in this paper. First, the clonal selection method is used to achieve the competition within population to select the individuals with high fitness values to reconstruct each population. The internal evolution of each population is completed during this process. Second, co-evolution operation is conducted to realize the information exchange among populations and this operation accords with the process of biological evolution. Finally, the evolutionary results are compared with the global best individual results, with a strategy called elitist preservation, to find out the individual who has the highest fitness value, that is, the result of clustering. The algorithms mentioned are tested in UCI datasets and artificial datasets, and are analyzed. In addition, the convergence and diversity of the proposed algorithm are tested in the paper and have been demonstrated above.

Although the proposed algorithm ICCE overcomes the shortcomings of FCM and ICSCA algorithms, and can get a better solution, there are some limitations with it such as have to specify the number of clusters which is a common problem of many clustering algorithms. And the time complexity of the proposed algorithm is relatively high, that makes it impossible to deal with big data. There are also some issues that are not discussed in the article such as the impact of the amount of populations. These issues will be discussed in future research.

Acknowledgments We would like to express our sincere appreciation to the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Natural Science Foundation of China, under Grants 61371201 and 61272279, the EU FP7 project (Grant No. 247619) on “NICaiA: Nature Inspired Computation and its Applications”, and the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

References

Agrawal R, Gehrke J, Gunopolos D (1998) Automatic subspace clustering of high dimensional data for data mining applications [C]. In: Proceedings of ACM SIGMOD international conference on management of data. ACM Press, New York, pp 94–105

Ahmad W, Narayanan A (2011) Population-based artificial immune system clustering algorithm [M]. In: Artificial immune systems, pp 348–360

Al-Muallim MT, El-Kouatly R (2010) Unsupervised classification using immune algorithm [J]. Int J Comput Appl 2(7):44–48

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure [C]. In: Proceedings of SIGMOD. ACM Press, New York, pp 49–60

Burnet MF (1957) A modification of Jernecs theory of antibody production using the concept of clonal election [J]. Austr J Sci 20(1):67–76

Chen YW, Huang L, Luo WM et al (2008) A dynamic clonal selection immune clustering algorithm[C]. In: 30th annual international conference of the IEEE. Engineering in Medicine and Biology Society, pp 1048–1051

De Castro LN, Von Zuben FJ (2000) The clonal selection algorithm with engineering applications. In: Proceedings of GECCO, workshop on artificial immune systems and their applications, pp 36–37

Deng ZH, Chung FL, Wang ST (2008) FRSDE: fast reduced set density estimator using minimal enclosing ball [J]. Pattern Recognit 41(4):1363–1372

Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters [J]. J Cybern 3(3):32–57

Du HF, Jiao LCH (2002) Clonal operator antibody clone algorithms. In: Proceedings of 2002 international conference on machine learning and cybernetics, vol 1, pp 506–510

Eghbal G, Mansoori (2013) GACH: a grid-based algorithm for hierarchical clustering of high-dimensional data. Soft Comput. doi:10.1007/s00500-013-1105-8

Ester M, Kriegel HP, Sander J, Xu XW (1996) A density-based algorithm for discovering clusters in large spatial databases with noise [C]. In: Proceedings of the 2nd international conference on knowledge discovering in databases and data mining. AAAI Press, pp 122–128

Ficici SG, Pollack JB (2000) A game-theoretic approach to the simple coevolutionary algorithm. PPSN, pp 467–476

Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976

Gao C, Pedrycz W, Miao DQ (2013) Rough subspace-based clustering ensemble for categorical data. Soft Comput 17:1643–1658

Girolami M, He C (2003) Probability density estimation from optimally condensed data samples [J]. Trans Pattern Anal Mach Intell 25(10):1253–1264

Guha S, Rastogi R, Shim K (1998) Cure: an efficient clustering algorithm for large database [C]. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data. ACM Press, New York, pp 73–84

Higham DJ, Kibble M (2004) A unified view of spectral clustering [R]. Department of Mathematics, University of Strathclyde, England

Hoppner F, Klawonn F, Kruse R, Runkler T (1999) Fuzzy cluster analysis [M]. Wiley, New York. http://archive.ics.uci.edu/ml/datasets.html

Jazen DH (1980) When is it co-evolution. Evolution 34:6118612

Jiao LC, Liu J, Zhong WC (2012) Coevolutionary computation and multiagent systems. WIT Press, UK

Jiang B, Wang N (2013) Cooperative bare-bone particle swarm optimization for data clustering. Soft Comput. doi:10.1007/s00500-013-1128-1

Kim J, Bentley PJ (2002) Towards an artificial immune system for network intrusion detection: an investigation of dynamic clonal selection. In: Proceedings of congress on evolutionary computation, pp 1015–1020

Kohonen T (1982) Self-organized formation of topologically correct feature maps [J]. Biol Cybern 43(1):59–69

Kotinis M (2013) Improving a multi-objective differential evolution optimizer using fuzzy adaptation and K-medoids clustering. Soft Comput. doi:10.1007/s00500-013-1086-7

Lee C, Zaïane O, Park H et al (2008) Clustering high dimensional data: a graph-based relaxed optimization approach [J]. Inf Sci 178:4501–4511

Lee D, Seung H (1999) Learning the parts of objects by nonnegative matrix factorization. Nature 401:788–791

Lin KW, Lin CH, Hsiao CY (2013) parallel and scalable CAST-based clustering algorithm on GPU. Soft Comput. doi:10.1007/s00500-013-1074-y

Lillesand T, Keifer R (1994) Remote sensing and image interpretation. Wiley, Hoboken

Liu RC, D HF, Jiao LC (2003) Immunity clonal strategies. In: ICCIMA, pp 290–295

Liu RC, Zhang XR, Yang N, Lei Q, Jiao LC (2012) Immunodomaince based clonal selection clustering algorithm. Appl. Soft Comput 12(1):302–312

Meila M, Xu L (2004) Multiway cuts and spectral clustering [R]. Department of Statistics, University of Washington, USA

Mézard M (2007) Where are the exemplars? Comput Sci 315(5814):949–951

Potter MA, De Jong KA (1994) A cooperative coevolutionary approach to function optimization. In: Proceedings of the international conference on evolutionary computation and the 3rd conference on parallel problem solving from nature, Jerusalem, Israel, pp 249–257

Potter MA, De Jong KA (1995) Evolving neural networks with collaborative species. In: Proceedings of the sixth international conference on genetic algorithms, pp 340–345

Potter MA, De Jong KA (1998) The coevolution of antibodies for concept learning. Evolut Comput 6(2):32–42

Potter MA, De Jong KA (2000) Cooperative co-evolutionary: an architecture for evolving co-adapted sub-components. Evolut Comput 8(1):1–29

Powers ST, Watson RA (2007) Preliminary investigations into the evolution of cooperative strategies in a minimally spatial model. In: GECCO, p 343

Rao MR (1971) Cluster analysis and mathematical programming. J Am Stat Assoc 66(335):622–626

Sheikholeslami G, Chatterjee S, Zhang A (1998) WaveCluster: a multi-resolution clustering approach for very large spatial databases [C]. In: Proceedings of the 24th VLDB conference. Morgan Kaufmann, pp 428–439

Wang W, Yang J, Muntz R. STING (1997) A statistical information grid approach to spatial data mining [C]. In: Proceedings of the 23rd VLDB conference. Morgan Kaufmann, pp 186–195

Zhang T, Ramakrishnan R, Livny M (1996) An efficient data clustering method for very large databases [C]. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data. ACM Press, New York, pp 103–114

Zhong YF, Zhang LP (2011) A new fuzzy clustering algorithm based on clonal selection for land cover classification [J]. Math Probl Eng 2011:1–21. doi:10.1155/2011/708459

Zhong YF, Zhang LP (2012) An adaptive artificial immune network for supervised classification of multi/hyper-spectral remote sensing imagery. J IEEE Trans Geosci Remote Sens 50(3):894–909