Integration of Particle Swarm Optimization and Immune Genetic Algorithm-Based Dynamic Clustering for Customer Clustering

R. J. Kuo
Department of Industrial Management, National Taiwan University of Science and Technology
No. 43, Section 4, Keelung Road, Taipei City 10607, Taiwan, R.O.C.
rjkuo@mail.ntust.edu.tw

S. H. Lin
Taiwan Semiconductor Manufacturing Company
No. 8, Li-Hsin Road, 6, Hsinchu Science Park, Hsinchu City 30078, Taiwan, R.O.C.
syuehua803@gmail.com

Zhen-Yao Chen
Department of Business Administration, De Lin Institute of Technology
No. 1, Ln. 380, Qingyun Road, Tucheng Dist, New Taipei City 25634, Taiwan, R.O.C.
keyzyc@gmail.com

Received 13 November 2013
Accepted 1 February 2015
Published 20 October 2015

This study intends to present a dynamic clustering (DC) approach based on particle swarm optimization (PSO) and immune genetic (IG) (DCPIG) algorithm, which is able to cluster the data into adequate clusters through data characteristics with pre-specified numbers of clusters. The proposed DCPIG algorithm is compared with three DC algorithms in the literature using Iris, Wine, Glass and Vowel benchmark data sets. The experiment results show that the DCPIG algorithm can achieve higher stability and accuracy than the other algorithms. In addition, the DCPIG algorithm is also applied to a real-world problem considering the customer clustering for a cyber flower shop. Lastly, we recommend different products and services to customers based on the clustering results.

Keywords: Cluster analysis; dynamic clustering; particle swarm optimization algorithm; immune genetic algorithm.

1. Introduction

Cluster analysis is a widely used technique. The goal of this technique is to partition a set of patterns into disjointed and homogeneous clusters. Basically, cluster analysis can group data with high similarity together and minimize the similarity among each cluster.

*Corresponding author
Through the process of reducing the complexity, a clustering algorithm also allows us to analyze the hidden commercial implications of the clustered data. This can be beneficial for business decision making. However, a challenge in data clustering is to determine the optimal number of clusters in the data set. Some efforts should be made to overcome this problem.\(^\text{20}\)

On the other hand, evolutionary algorithms (EAs), which allow us to find the global solution for a given problem, are heuristic and stochastic search procedures based on the mechanics of natural selection, genetics, and evolution.\(^\text{50}\) They include the genetic algorithm (GA), particle swarm optimization (PSO), and artificial immune system (AIS) algorithm. Thus, this study attempts to propose a dynamic clustering (DC) approach based on PSO and the immune genetic (IG) (DCPIG) algorithm that possesses automatic clustering capability. In other words, the proposed DCPIG algorithm does not require a pre-specified number of clusters. It can automatically cluster data into adequate clusters by examining the features of the data. In order to verify the DCPIG algorithm, this study employs four benchmark data sets: Iris, Wine, Glass, and Vowel. The computational results are compared with other algorithms including the DC approach based on PSO and GA (DCPG),\(^\text{37}\) DC approach based on binary-PSO (DCPSO),\(^\text{52}\) and DC approach based on GA (DCGA)\(^\text{45}\) algorithms.

In addition, the Internet is well known for the high degree of interaction with customers. The increasing popularity of the Internet provides enterprises with another potential marketing channel. Thus, if an enterprise wants to preserve its competitive advantage in the electronic commerce environment, well-established customer relationship management (CRM) is extremely important. Furthermore, to extract valuable information with commercial implications among the huge and complex transactional data collected from the Internet, the cluster analysis algorithm has become a useful tool to support business decision making. Therefore, we further apply the proposed DCPIG algorithm to the real-world problem of a cyber flower shop located in Taiwan. The purpose is to cluster the customers according to customer values in the transactional database. Basically, the results can be applied as a base for CRM. This case company can therefore recommend customized products or services to different customer segments.

The remainder of this study is organized as follows. Section 2 briefly presents the necessary background by discussing cluster analysis, EAs, integrated EAs, and CRM. Section 3 explains the proposed DCPIG algorithm. Next, Sections 4 and 5 discuss the analysis of experimental results using four benchmark data sets and the model evaluation results for a real-world problem, respectively. Finally, concluding remarks are made in Section 6.

2. Literature Review

This section briefly discusses the related background associated to this study, including cluster analysis, evolutionary algorithms (EAs), integrated EAs, and customer relationship management (CRM).
2.1. Cluster analysis

Cluster analysis partitions data into a certain number of clusters. Most researchers describe a cluster by considering internal homogeneity and external separation, i.e., patterns in the same cluster should be similar to each other, while patterns in different clusters should not. In the past, many clustering methods have been proposed from Ref. 66. Generally, these methods can be broadly divided into two classes: hierarchical and partitional. Hierarchical clustering proceeds successively by merging smaller clusters into larger ones or by splitting larger clusters. On the other hand, partitional clustering attempts to directly decompose the data set into several disjointed clusters based on some criteria. The most common criterion adopted is to minimize dissimilarity within each cluster and maximize the dissimilarity of different clusters. However, most hierarchical and partitional clustering methods have a drawback in that the number of clusters needs to be specified a priori. Thus, dynamically determining the optimal number of clusters for a data set is a challenging work, since a priori knowledge of the data is not always available.

An unsupervised dynamic clustering algorithm based on the combination of fuzzy C-means and fuzzy maximum likelihood estimation from Ref. 17 had proposed. Further, an algorithm based on fuzzy clustering to dynamically determine the number of clusters for a data set from Ref. 48 had proposed. Furthermore, an alternative approach (i.e., SYNERACT) to ISODATA combines K-means with hierarchical descending approaches. In this approach, it is also unnecessary to specify the number of clusters. Based on K-means, other DC algorithms have also been developed, such as X-means and G-means.

Additionally, an artificial neural network (ANN) can be applied for DC. ANNs are systems that have been derived through models of neurophysiology. In general, they consist of the collection of simple nonlinear computing elements whose inputs and outputs are linked to form networks. ANN-based clustering has been dominated by the self-organizing feature map (SOM) and adaptive resonance theory (ART). Some important representative algorithms include SOM, ART1, ART2, and fuzzy ART.

Moreover, Ref. 40 applied an evolution strategy to dynamically cluster a data set. The evolution strategy they proposed implements variable length genomes to search for both the centroids and numbers of clusters. A DC approach based on GA (DCGA) algorithm had proposed from Ref. 45 that can automatically finds the proper number of clusters, and the recovery of their algorithm surpasses K-means with the Iris data set. This DCGA algorithm is suitable for clustering the data with compact spherical clusters. It can be used in two ways. One is the user-controlled clustering, where the user may control the result of clustering by varying the values of the parameter. The other is the automatic clustering, where a heuristic strategy is applied to find a good clustering.

In addition, Ref. 38 presented an ant colony-based clustering method for case clustering. Also, Ref. 52 proposed a new DC approach based on the binary-PSO (DCPSO) algorithm and applied it to image segmentation. The DCPSO algorithm automatically determines the “optimum” number of clusters and simultaneously clusters the data set with
minimal user interference. It starts by partitioning the data set into a relatively large number of clusters to reduce the effects of initial conditions in which the best number of clusters is selected through binary-PSO algorithm. The centers of the chosen clusters are then refined via the K-means clustering algorithm. Further, the DCPSO algorithm was applied on both synthetic and natural images. The experiments conducted show that the proposed DCPSO algorithm generally found the “optimum” number of clusters on the tested images. Afterward, Ref. 55 applied PSO algorithm for partitional clustering. Further, Ref. 53 presented a modified PSO algorithm for automatic image clustering.

After that, based on PSO and GA algorithms, Ref. 37 proposes the DCPG algorithm to solve the problem of setting the number of clusters in advance and find out the suitable number of clusters based on the features of data. As for the empirical study, the DCPG algorithm was applied to cluster analysis for the bills of material (BOM) provided by Advantech Company. Further, the DCPG algorithm found a suitable number of clusters, clustering results, product clusters, and shared materials.

2.2. Evolutionary algorithms (EAs)

Evolutionary algorithms (EAs) are more flexible than classical ones as they do not require optimizing functions to have properties such as differentiability and continuity. They have a better ability to escape from local optima with robust global searching and offer an efficient tool to optimize the model structure. The core procedure in EAs is repeating the generational steps of selection, reproduction, and evaluation. These characteristics of EAs, as well as other supplementary benefits such as ease of implementation, parallelism, no requirement for a differentiable or continuous objective function, etc., make them an attractive choice for general optimization problems. Next, we briefly introduce some EAs that imitate certain natural principles such as the PSO and IG algorithms in the following subsections.

2.2.1. Particle swarm optimization (PSO) algorithm

Recent developments in EAs have introduced quite a few natural processes and one among them is the PSO algorithm. This algorithm has evolved as an important branch of stochastic techniques that explores the search space for optimization. The PSO algorithm has been applied to simulate the graceful motion of bird swarms as a part of a socio-cognitive study. In further detail, each particle has a position vector and a velocity vector in the n-dimensional search spaces, and thus can be represented as \( X_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t)) \) and \( V_i(t) = (v_{i1}(t), v_{i2}(t), \ldots, v_{in}(t)) \) respectively. Each particle also keeps track of its best position value, and the best position encountered by each particle is represented as \( P_i(t) = (p_{i1}(t), p_{i2}(t), \ldots, p_{in}(t)) \). The best particle among all particles found so far at time \( t \) is represented as \( P_g(t) = (p_{g1}(t), p_{g2}(t), \ldots, p_{gn}(t)) \). The algorithm can then be applied using the following equations:

\[
\begin{align*}
\dot{v}_{in}(t+1) &= k \cdot v_{in}(t) + c_1 \cdot r_{i1}(t) \cdot (p_{in}(t) - x_{in}(t)) + c_2 \cdot r_{i2}(t) \cdot (p_{g'in}(t) - x_{in}(t)), \\
x_{in}(t+1) &= x_{in}(t) + v_{in}(t+1),
\end{align*}
\]
where \( r_{1n}(t) \) and \( r_{2n}(t) \) are random numbers uniformly distributed in the interval (0, 1), \( c_1 \) and \( c_2 \) are acceleration constants, and \( k \) denotes the inertia weight. The \( k \) is a user-defined parameter that controls \( c_1 \) and \( c_2 \), which are the previous values of particle velocities on its current ones. The \( c_1 \cdot r_{1n}(t) \cdot (p_{in}(t) - x_{in}(t)) \) component in Eq. (1), referred to as the cognitive component, represents the distance that a particle is from the best solution \( (p_{in}(t)) \) found by itself. The \( c_2 \cdot r_{2n}(t) \cdot (p_{gn}(t) - x_{in}(t)) \) component in Eq. (1), referred to as the social component, represents the distance that a particle is from the best solution found for its neighborhood. Eq. (1) is used to calculate the new velocity of the particle according to its previous velocity and the distances of its current position from its own best historical position and the collaborative effect of particles. Knowledge (information) is shared through the cooperation between all particles. Then, the particle updates the new position using Eq. (2).^{39}

2.2.2. Immune genetic (IG) algorithm

Reference 23 originally developed the genetic algorithm (GA) in the 1960s, and it has been further described by Ref. 19. GA is a stochastic global search technique that solves problems by imitating processes observed during natural evolution. Based on the survival and reproduction of the fittest, GA continually exploits new and better solutions without pre-assumptions, such as continuity and uni-modality. GA simulates biological evolutionary behavior. GA adopts the concept of species extinction and finds the approximate optimal solution after the process of coding, decoding and constant operation (reproduction, crossover, and mutation). Later, based on the principle of the biological immune mechanism, Ref. 27 used the adaptive immune response characteristics of the principle of learning derived human information processing. They then integrated GA with crossover and mutation operations to develop a new algorithm to search for the optimal solution. This is referred to as the immune genetic (IG) algorithm. In addition, Ref. 51 applied an immune system for data clustering which does not require any user specified number of clusters.

For the IG algorithm,^{27} the first step is to understand the objective function and all related restrictions clearly, and then decide how to encode the feasible solution. Reference 27 presented the IG algorithm procedures as follows:

1. **Initial population generation**
   - Randomly generate a number of fixed-size feasible solutions in the solution space, and use them as the initial population to search for the optimal solution.

2. **Fitness value**
   - Calculate the fitness value of various ethnic groups within the randomly generated initial population. The fitness values will be the basis of the levels for the antibodies. It helps to find antibodies that are suitable to be selected to copy to the next epoch.

3. **Clone**
   - According to the adaptation level of each antibody, select some good antibodies for cloning. The clone method is similar to the roulette method. Basically, an antibody is
randomly selected and its clone is implemented for the selected antibodies. This study uses the roulette selection method to clone antibodies. This method attains better performance and fitness values.

(4) **Crossover**

In order to retain some of the characteristics of antibodies, the crossover procedure is only implemented with a certain probability. Thus, it is necessary to determine the crossover rate. The crossover is implemented as follows:

\[ X_c = \text{Uniform} (0, 1) X_A + (1 - \text{Uniform} (0, 1) X_B) \]  

(3)

(5) **Mutation**

Mutation is a process to avoid losing opportunities to generate new information for the antibodies during crossover, since crossover may fall into a local optimal solution and mutations can help the solution escape from the current space. This study adds a small value to the antibodies as a disturbance to help the antibodies mutate, as shown in the following equation.

\[ X'_A = X_A + \text{rand} \times N(0, 1) \]  

(4)

(6) **Size of the memory cell**

Due to the characteristics of the immune system, it recollects the antibody that is effective in resisting the antigen, and then makes it react quickly. In this study, the memory cell contains the best feasible solutions obtained from each iteration.

2.3. **Integrated EAs**

From the description of PSO and IG algorithms, the difference between the two is that PSO lacks crossover and mutation and easily falls into the local optimal solution. However, PSO can memorize the global best solution and affect the movement of other particles. These characteristics result in quick convergence and falling into the local optimal solution. In addition, IG algorithm shares information through chromosomes. Some studies have suggested that hybrids of PSO and GA algorithms can obtain better solutions.

Reference 57 proposed the PSO-GA algorithm. This algorithm uses the optimal solution of PSO as the initial parent of GA. Later, Ref. 14 put forward GA-PSO and applied it to an ANN. They substituted an operator of GA into PSO and updated the position to achieve certain advantages when using GA-PSO. The aforementioned two methods use GA after PSO, and another hybrid method features the simultaneous operation of PSO and GA for production of the next iterative individual. Reference 30 suggested the GA-PSO algorithm. This method divides the randomly generated parents into the one with better fitness value and the one with worse fitness value. GA is carried out for the one with better fitness value, and PSO is performed for the one with worse fitness value. Then, the result calculated from GA is used to adjust PSO individuals and update the action. The GA-PSO algorithm integrates the individual evolution of GA and the self-improvement of PSO algorithm.
In addition to combining PSO with the operators of GA, elitist selection can be adopted to increase algorithm’s efficiency. Accordingly, Ref. 29 proposed a hybrid of GA and PSO (HGAPSO) algorithm, which discards particles with worse fitness value, and refers to the particles with better fitness value as elitist. The method can increase the influence of the particles with better fitness value and convergence speed. Consequently, Ref. 35 proposed using the hybrid of PSO and GA based optimization (HPSGO) algorithm to improve performance of a radial basis function neural network (RBFnn). In the case of having limitation on local optimal solution, GA can be utilized to change the chromosomes and escape the local optimal solution to quickly attain the global optimal solution. The computational results indicated that the HPSGO algorithm is superior to PSO or GA. Reference 36 proposed the hybrid of GA and PSO algorithms (HGAPSOA) clustering algorithm for order clustering. Evaluation results using data provided by an IPC manufacturer also show that the proposed HGAPSOA is more accurate than the GA-based and PSO-based algorithms proposed by previous studies.

When incorporated with EAs, the immune system can improve the search ability during the evolutionary process. Reference 16 proposed a hybrid artificial immune network (HaiNet) with swarm learning, which is inspired not only by biological immune system but also by the group behaviors of PSO. The policy of swarm learning in PSO is introduced into the hyper-mutation operation of antibodies in AIS. Moreover, Ref. 42 proposed an efficient artificial immune network (EaiNet) algorithm for function optimization with elite learning. The spirit of the EaiNet algorithm lies in mutation operations, which not only emphasizes the self-evolution of antibodies but also absorbs the elite learning mechanism of PSO. The other spirit of EaiNet algorithm is that it selectively tackles the global elite antibody and other antibodies during mutation operation. For the global elite antibody, elite-keeping and self-learning are arranged based on affinity. Also, Ref. 67 integrated the AIS and a chaotic operator into the classical PSO to form a novel chaotic immune PSO (CIPSO) algorithm to solve the model.

2.4. Market segmentation

Customer relationship management (CRM) is a derivative of the earlier American term “contact management” (during the 1980s). From the report of Ref. 60, one can find out that extended functions of “contact management” include customer data collection, as well as the gathering and application of useful information. The benefits of CRM implementation not only assist enterprises in locating business opportunities, but also improve competitive advantage through lowering cost and gaining higher customer value (CV) in comparison with the competition. Basically, the idea of CRM is based upon relationship marketing. Reference 59 went further to suggest that the integration of business applications and data sources can be realized through the recent developments in Web service solutions, which can then be further incorporated into CRM approaches. To understand a CRM system from the aspect of marketing, its ultimate target involves how to fit the customer’s requirements and maintain the long-term customer relationship.
Fig. 1. The flowchart for the proposed DCPIG algorithm.

Managers are currently becoming increasingly dependent on the Internet in order to create business value. Prior research has analyzed the importance of product characteristics, such as frequency of purchase, value position, and appropriateness of sales on the Internet when adopting the web as a marketing channel. The fast growing Internet network and communication technologies provide services such as e-mail, online interaction, and personal websites. These services have become important resource for companies attempting to achieve a one-on-one relationship, CV analysis, and mass customization. Enterprises are required to evaluate the value of their customers, segment customers based on CV and develop strategies for every customer segment to acquire and retain profitable customers. As enterprises successfully improve the lifetime
value of customers, they will improve their investment returns, enhance customer loyalty, increase the profits from the existing customers, improve the value of their database, and so on.\(^\text{43}\)

3. Methodology

Most clustering algorithms require a pre-specified number of clusters. However, it is hard to obtain the number of clusters in practical applications. Thus, this study proposes an integrated EA for DC. The basic idea comes from the DCPSO algorithm proposed by Ref. 52. It applies PSO to update the number of clusters dynamically. However, though DCPSO algorithm can find the acceptable solution in a reasonable time, it can easily get stuck to the local optimum. Thus, this study presents a DC approach based on PSO and the immune genetic (IG) (DCPIG) algorithm, which possesses automatic clustering capability and is able to provide fast convergence and accuracy for cluster analysis. Figure 1 illustrates the flowchart of the proposed DCPIG algorithm in this research.

3.1. The proposed DCPIG algorithm

The evolutionary procedure of the DCPIG algorithm is as follows:

Step 1. Set up the parameters including the population size, inertia weight \((W)\), maximum velocity \((V_{\text{max}})\), learning factors \((c_1, c_2)\), the maximum number of clusters \((N_c)\), crossover rate, and mutation rate.

Step 2. Randomly generate \(N_c\) cluster centers \((M)\).

Step 3. Randomly generate the initial position \((X_{id})\) and initial velocity \((V_{id})\) of individual particles in the population for the DCPSO algorithm, where

\[
X_{id} = (x_{i1}, \ldots, x_{ik}, \ldots, x_{iN_c})
\]

In which, \(x_{ik} \sim U(0, 1)\) and \(N_c\) is in the range of \([0, 1]\). If \(x_{ik} = 1\), it represents that the \(i\)th particle belongs to the \(k\)th cluster. If \(x_{ik} = 0\), it represents that the \(i\)th particle does not belong to the \(k\)th cluster.

Step 4. Calculate the fitness values of all particles to measure their performance. The fitness value calculation for each particle is based on the following steps:

1. Partition data according to the centroids shown in the particle by assigning each data point to the closest (in terms of the Euclidean distance) cluster.

2. Calculate the fitness value of each particle based on Eq. (6):

\[
\text{fitness value} = \sum_{k=1}^{N_c} \left( \sum_{Z \in z_k} \|Z - M_k\| \right).
\]

Step 5. Select \(P_{id}\) and \(P_{gd}\).

Step 6. Adjust the velocity \((V_{id})\) and position \((X_{id})\) of individual particles through the updating rule of the binary-PSO algorithm as follows:

\[
V_{id}^{new} = w \times V_{id}^{old} + c_1 \times \text{rand}_{id} \times (P_{id} - X_{id}^{old}) + c_2 \times \text{rand}_{2d} \times (P_{gd} - X_{id}^{old}),
\]
sigmoid(V_{id}^{\text{new}}) = \frac{1}{1 + e^{-V_{id}^{\text{new}}}}, \quad (8)

X_{id}^{\text{new}} = \begin{cases} 1, & \text{if } \text{rand}(\cdot) < \text{sigmoid}(V_{id}^{\text{new}}) \\ 0, & \text{else} \end{cases} \quad (9)

Step 7. With the updated population generated from Step 6, take the following procedures:

1. Reproduce the initial population and then generate population 1. In addition, apply a crossover operator to population 1 with P_{id} and P_{gd} selected from Step 5.
2. Apply a mutation operator to the selected P_{gd} to generate population 2. Furthermore, execute the IG algorithm with steps described below:
   a. Calculate the affinity value (i.e., $\alpha$) of each particle in the generated population after updating.
   b. Calculate the concentrate value (i.e., $\psi$) of each antibody in the generated population after updating.
   c. If the value of $\psi$ is larger than the average concentrate value of all antibodies in the generated population after updated, it will generate the inhibitive cell (i.e., antibody) otherwise it will generate the memory cell.
   d. Implement reproduction and inhibition for the generated antibodies in this population, and then generate a new population.
   e. Repeat the above steps until a pre-specified number of epochs has been satisfied, otherwise return to procedure (b).

Step 8. Combine population 1 and population 2, and calculate the fitness values of individual particles in the new population.

Step 9. Apply the elitist selection\(^{29}\) procedure to populations 1 and 2, respectively. Then it will generate new populations 1 and 2 for the next iteration.

Step 10. Return to Step 4 until the pre-specified number of clusters is satisfied.

Step 11. Adjust $M_{jk}$ through the position of extreme values in the population, and then apply the K-means algorithm to refine the cluster centers. The detailed procedures are as follows:

1. Take the relative point of center for the particle whose position is 1 in relation to the initial center of the cluster in the K-means algorithm.
2. As for the sum of positions from the extreme values in the population, it can be regarded as the initial number of clusters.

Step 12. Randomly select $M_{rk}$ and calculate $M$ through Eq. (10) as follows:

$$M = M_{rk} \cup M_{jk},$$ \quad (10)

where $M$ would replace the mapping center of the extreme values in the population whose position is zero.

Step 13. The DCPIG algorithm will not stop returning to Step 4 until a pre-specified number of epochs has been satisfied.
In general, the clonal selection principle has been utilized to design immunological algorithms with self-organizing and learning capability for solving combinatorial optimization problems. AIS, which was inspired by the theory of immunology, is one of the evolutionary techniques developed recently. Furthermore, immunology is the scientific discipline that studies the response of AISs when a non-self antigenic pattern is recognized by antibodies.

By integrating the PSO and IG algorithms mentioned above, it is promising to conclude that the proposed DCPIG algorithm can significantly increase the diversity of population-based solutions in the process of evolution, and thus increase the possibility of solving the global optimal solution. Then, the best solution can be obtained and used in the DCPIG algorithm to solve the problem of automatic clustering.

4. Experimental Analysis

In general, evolutionary computations (ECs) inherit the principles of biological evolution. This is stochastic in nature and stronger as compared to traditional optimization methods. Further, considering the drawbacks of traditional optimization techniques, attempts are being made to solve the optimization problems by using meta-heuristics, which are mostly nature inspired such as PSO, GA, AIS, GA, and differential evolution (DE) algorithms. Moreover, Ref. 31 have pointed out that “EAs are capable of finding the near-optimal solution to the numerical real-valued test problems for which exact and analytical methods do not produce the optimal solution within a reasonable computation time”, especially when the global minimum is surrounded by many local minima. After that, the EAs are based on the EC method and have significantly shown their outstanding ability in solving complex nonlinear optimization problems.

In addition, owing to clustering can be regarded as a category of optimization problems that uses EAs such as GA, PSO or AIS algorithms. Therefore, when we tried to develop a better dynamic clustering algorithm to cluster data into adequate number of clusters by examining data features and perform excellent clustering results, several algorithms in literatures are appropriate as the basis of comparisons. Subsequently, this study employs four benchmark data sets: Iris, Wine, Vowel, and Glass, with known clustering results to assess the proposed DCPIG algorithm. The results are compared with those of DCPG, DCPSO, and DCGA algorithms.

On the other hand, a superior clustering algorithm is expected to possess minimum variance for intra-clusters but significant variance for inter-clusters (i.e., between two clusters). Therefore, the ratio of average intra-cluster distance to minimum inter-cluster distance (i.e., validity index (VI) value) is adopted. We implemented 30-time trials for each algorithm on each data set. Then, we calculated the averaged value of 30-times trials for the purpose of comparison. Furthermore, the VI and Index values can thus be employed as the measurement.


4.1. Data sets

This study applies the benchmark data sets from the database provided by the Department of Information and Computer Science at UCI (http://www.ics.uci.edu/~mlearn/databases). They include Iris, Wine, Glass, and Vowel. The data tuples, feature dimensions, and cluster numbers are illustrated in Table 1.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Data Tuples</th>
<th>Feature Dimension</th>
<th>Cluster #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Vowel</td>
<td>990</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

4.2. Parameter setup

In the proposed DCPIG algorithm, four parameters (i.e., inertia weight, learning factor, crossover rate, and mutation rate), which have significant impact on computation results, are analyzed. Subsequently, this section adopts Glass data set to apply trial analysis using Taguchi experiment design and the Taguchi trials were configured in an L9(3^4) orthogonal array for the DCPIG algorithm after the experiment was implemented thirty times. Meanwhile, the statistical software MINITAB 14 was used in the analysis of parameter design for algorithm, where the signal-to-noise (S/N) ratio is used to evaluate the stability of system quality in the experiment. Finally, Table 2 presents the parameter setup for the proposed DCPIG algorithm.

Additionally, the maximum number of clusters, which should not exceed the square root of data tuples according to Ref. 68, should also be determined for the four data sets before implementation. Thus, the maximum number of clusters for the Iris and Wine data sets is 13. For the Glass and Vowel data sets, they are 15 and 30, respectively.

<table>
<thead>
<tr>
<th>The Description of Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The total number of training (epochs)</td>
<td>500</td>
</tr>
<tr>
<td>Population size</td>
<td>20</td>
</tr>
<tr>
<td><strong>PSO algorithm part:</strong></td>
<td></td>
</tr>
<tr>
<td>Inertia weight (W)</td>
<td>[1.2, 0.9]</td>
</tr>
<tr>
<td>Learning factor (c1, c2)</td>
<td>c1 = [0.35, 2.4], c2 = [2.4, 3.5]</td>
</tr>
<tr>
<td>The maximum velocity of each particle (Vmax)</td>
<td>3</td>
</tr>
<tr>
<td><strong>IG algorithm part:</strong></td>
<td></td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.6</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.05</td>
</tr>
<tr>
<td>Affinity rate (α)</td>
<td>0.7</td>
</tr>
<tr>
<td>Concentrate rate (ψ)</td>
<td>0.75</td>
</tr>
</tbody>
</table>
4.3. Data preprocessing

This study uses distance as the measuring index for the clustering results. As for the data set, as the variance between feature values is significant, such significant degrees of data fluctuation would produce deviation in the distance measurement. Thus, if the raw data of the data set is used for cluster analysis, it is expected to have an inferior impact on the clustering results. Thus, the raw data are preprocessed through normalization, so the data has a certain standard when measuring different feature values. The normalization formula is shown as Eq. (11):

\[ x_i = \lambda_1 + (\lambda_2 - \lambda_1) \left( \frac{z_i - z_i^{\text{min}}}{z_i^{\text{max}} - z_i^{\text{min}}} \right), \]  

where \( x_i \) is the normalized data, \( z_i \) is the raw data, \( \lambda_1 \) is the minimum normalized value, \( \lambda_2 \) is the maximum normalized value, \( z_i^{\text{min}} \) is the minimum value from raw data, and \( z_i^{\text{max}} \) is the maximum value from the raw data.

4.4. Evaluation principle

One approach to determine the optimal number of clusters is to execute the clustering algorithm multiple times, each time with a different number of clusters, and validate the clustered data set with a cluster value index. Thus, the current study uses the modified validity index (VI) as the evaluation index, as seen in Eq. (12):

\[ VI = (c \times N(0,1) + 1) \times \frac{\text{intra}}{\text{inter}}, \]

where \((c \times N(0,1) + 1)\) is a punishing value to prevent data convergence from obtaining too small of a number of clusters. With this adjusted value, the simplified data will not converge into just two clusters but move toward a more adequate number of clusters. Among which, \( c \) is a constant and is set as 30 in this study. \( N(0,1) \) is the Gaussian function of the resolved number of clusters. The \( \text{intra} \) is the averaged intra-cluster distance which is shown as Eq. (13):

\[ \text{intra} = \frac{1}{N_p} \sum_{k=1}^{K} \sum_{u \in C_k} \|u - m_k\|^2. \]  

Its purpose is to calculate the intensity of intra-clusters; the approach is to calculate the Euclidian distance of the data point and the center of the cluster, then sum up all the shortest distances from each data point to the center of the cluster and finally divide by the total data tuples \((N_p)\). The smaller (larger) the \( \text{intra} \) value is, the better (worse) the clustering efficiency of the algorithm is considered to be. Lastly, the \( \text{inter} \) value is the distance between two clusters and is shown as Eq. (14):

\[ \text{inter} = \min \left\{ \|m_k - m_{kk}\|^2 \right\} \quad \forall \ k = 1, 2,\ldots, k - 1 \text{ and } kk = k + 1,\ldots, K. \]  

The \( \text{inter} \) value is aimed to calculate the minimum inter-cluster distance. Once the distance from a center to all other centers of clusters is determined, the minimum inter-cluster distance is then calculated.
The expected clustering result is a small intra-cluster distance with large inter-cluster distance. However, it is hard to identify the quality of the clustering result using the $VI$ value. Thus, the sum of squares within ($SSW$) and the sum of squares between ($SSB$) are calculated in each iteration as well. Moreover, the other $Index$ value is used to determine the number of clusters. Formulae associated with the above calculations are shown in the followings from Eqs. (15) to (17):

$$SSW = \sum \sum (X_{ij} - \bar{X}_{k})^2,$$

where $\bar{X}_{k}$ is the center of the cluster.

$$SSB = \sum N_i (\bar{X}_{k} - \bar{X})^2,$$

where $N_i$ is the data tuples of cluster, $\bar{X}_{k}$ is the center of the cluster, and $\bar{X}$ is the average of all data. Therefore, the $Index$ formula is shown as Eq. (17):

$$Index = \frac{SSW}{SSB}.$$


4.5. Results analysis

This study uses the number of clusters, the $VI$ value, and the $Index$ value as the evaluation criteria. In addition, since we generate the initial solution of the clustering algorithm randomly, the results will be different every time. Thus, this study calculates the averaged value and standard deviation (SD) of the 30-time trials for assessment. Basically, if the number of clusters is closer to the known number of clusters for the benchmark data set, this indicates that the clustering efficiency is better. Table 3 presents the results of the comparison of all clustering algorithms.

For the Iris data set, Table 3 indicates that the proposed DCPIG algorithm has 3 clusters, which is exactly equal to the known number of clusters. Also, the SD and $Index$ values for the DCPIG algorithm are the smallest among all the algorithms in the Iris data set. This means that DCPIG algorithm is the most stable method. In addition, the average number of clusters (i.e., 3.56666) for the DCPIG algorithm is the closest to the known number of clusters (i.e., 3) among all algorithms for the Wine data set. Moreover, the $VI$ and $Index$ values for the DCPIG algorithm are the smallest among all algorithms. For the other two data sets, Glass and Vowel, the situation is also similar. Thus, we can conclude that the proposed DCPIG algorithm can provide the best number of clusters and is more stable than the other clustering algorithms.

4.6. Convergence analysis

Figures 2–5 illustrate the convergence curves for all algorithms for the Iris, Wine, Glass, and Vowel data sets. Figure 2 shows that the DCPIG algorithm possesses the best convergence for the Iris data set. The DCGA algorithm cannot converge very smoothly, but the DCPSO algorithm can. Additionally, Fig. 3 shows that the DCPIG algorithm converges to the optimal solution at around 20 iterations for the Wine data set while the
Table 3. The clustering results among all algorithms.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Criteria</th>
<th>DCPIG (Kuo et al., 2012)</th>
<th>DCPG (Omran et al., 2006)</th>
<th>DCPSO (Lin &amp; Shiueng, 2001)</th>
<th>DCGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Cluster #</td>
<td>3</td>
<td>3</td>
<td>3.066667</td>
<td>3.033333</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0</td>
<td>0</td>
<td>0.249444</td>
<td>0.179505</td>
</tr>
<tr>
<td></td>
<td>VI</td>
<td>0.34144</td>
<td>0.35869</td>
<td>0.371549</td>
<td>0.374186</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.01249</td>
<td>0.01435</td>
<td>0.022731</td>
<td>0.024688</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>0.20049</td>
<td>0.206311</td>
<td>0.208975</td>
<td>0.20607</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0</td>
<td>0.002203</td>
<td>0.029518</td>
<td>0.00659</td>
</tr>
<tr>
<td>Wine</td>
<td>Cluster #</td>
<td>3.56666</td>
<td>3.766667</td>
<td>3.63333</td>
<td>3.933333</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.495536</td>
<td>0.422953</td>
<td>0.60461</td>
<td>0.628932</td>
</tr>
<tr>
<td></td>
<td>VI</td>
<td>0.53629</td>
<td>0.58139</td>
<td>0.596555</td>
<td>0.615231</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.02661</td>
<td>0.02386</td>
<td>0.035552</td>
<td>0.045821</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>0.94715</td>
<td>0.96284</td>
<td>0.969219</td>
<td>0.978729</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.07003</td>
<td>0.05652</td>
<td>0.07855</td>
<td>0.19404</td>
</tr>
<tr>
<td>Glass</td>
<td>Cluster #</td>
<td>6</td>
<td>5.93333</td>
<td>5.43333</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.516398</td>
<td>0.77172</td>
<td>0.61554</td>
<td>1.469694</td>
</tr>
<tr>
<td></td>
<td>VI</td>
<td>0.43285</td>
<td>0.46991</td>
<td>0.538998</td>
<td>0.44367</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.04876</td>
<td>0.048367</td>
<td>0.084971</td>
<td>0.04233</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>0.59152</td>
<td>0.57861</td>
<td>0.635631</td>
<td>0.676888</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.085106</td>
<td>0.10041</td>
<td>0.111929</td>
<td>0.112558</td>
</tr>
<tr>
<td>Vowel</td>
<td>Cluster #</td>
<td>11.1</td>
<td>11.66667</td>
<td>8.3</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>2.454248</td>
<td>2.573368</td>
<td>2.019076</td>
<td>3.723797</td>
</tr>
<tr>
<td></td>
<td>VI</td>
<td>0.76784</td>
<td>0.781935</td>
<td>0.74063</td>
<td>0.788056</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.03476</td>
<td>0.03316</td>
<td>0.042245</td>
<td>0.034336</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>0.74984</td>
<td>0.719185</td>
<td>0.930129</td>
<td>0.61257</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.13251</td>
<td>0.118935</td>
<td>0.165263</td>
<td>0.167886</td>
</tr>
</tbody>
</table>

Other algorithms need a greater number of iterations. Moreover, Fig. 4 also shows that the DCPIG algorithm possesses the best convergence among all algorithms for the Glass data set. Lastly, Fig. 5 shows that the DCPIG algorithm converges to the optimal solution at around 80 iterations when handling the complex Vowel data set. As for the unstable convergence of the DCGA algorithm, it could be caused by the mutation operator, which may destroy the good solution.
Fig. 2. (Color online) The convergence curves among all algorithms for the Iris data set.

Fig. 3. (Color online) The convergence curves among all algorithms for the Wine data set.
5. Model Evaluation Results

This section will apply the proposed DCPIG algorithm to cluster the transaction data provided by a case company to some segments. The case company is a business-to-consumer (B to C) cyber flower shop located in Taiwan. The overall transaction data for the cyber flower shop is from 2006/1/1 to 2006/12/31. The data are calculated and pre-processed through the RFM (recency, frequency, and monetary) rule first. Thus, there are three input attributes for each piece of data. Then the proposed DCPIG algorithm is applied to segment the customers. The goal is to categorize customers with similar

Fig. 4. (Color online) The convergence curves among all algorithms for the Glass data set.

Fig. 5. (Color online) The convergence curves among all algorithms for the Vowel data set.

1550019-17
characteristics into one cluster, so the enterprise can develop corresponding marketing strategy based on each customer cluster.

5.1. Data preprocessing

Reference 49 noted that enterprises should understand RFM variables to determine suitable marketing strategy. The RFM analysis technique identifies customer behavior according to three variables as follows:

1. Recency \( (C_R) \) of the last purchase refers to the interval between the time that the latest purchase happened and the present.
2. Frequency \( (C_F) \) of purchases refers to the number of transactions in a particular period.
3. Monetary value \( (C_M) \) of a purchase refers to the amount of a particular order.

Basically, customer lifetime value or loyalty can be evaluated in terms of RFM variables by integrating the rate of each cluster (i.e., \( C'_j \)), that is:

\[
C'_j = w_R C'_R + w_F C'_F + w_M C'_M, \tag{18}
\]

where \( w_R, w_F, \) and \( w_M \) are the relative importance of the RFM variables. The RFM rule is generally acknowledged as the most popular customer value analytical method at present.

The detailed procedure for data transportation is elaborated as follows:

(1) **Data integration**
   Combining the profile and transactional data based on the shopping interval of an individual customer to resolve the value of the CV.

(2) **Data transformation**
   Transform or alter if the data does not correspond with the right format or if incorrect information exists.

(3) **Data filtering**
   In practical cases, clustering analysis only aims at the RFM attributions from the customer transactional database. Therefore, this study only extracts the required shopping date and total transaction amount of individual customer.

(4) **Differentiating the date interval**
   In this study, the date interval of the customer transaction data is set to one year. Thus, if a customer has no transaction record during one year, then it is marked as a customer lost.

(5) **Calculating the values of RFM attributions**
   Calculate the values of RFM attributions for individual customers from the transactional database. For instance, a customer had two transaction records; one on 2006/11/1, with an order value of $900, and one on 2006/11/16, with an order value of $1100, respectively. For this customer, the R-value is 15 days as the date interval between the two transactions, the F-value is 2 times, and the M-value is the total amount of the two orders, $2000.
Computing the PR of RFM values
In general, the value of RFM attributions is marked as 1 (low CV) to 5 (high CV) for individual customers. In this study, the PR value that between zero and one is used to rank the values of RFM attributions for each customer. For example, the values of R, F, and M are 0.5, 0.8, and 0.6 for one particular customer. Thus, the CV vectors are 0.5, 0.8, and 0.6.

Calculating the CV Index
The monetary situations of individual customers are the primary evaluation in this study. Meanwhile, we use the SD of each transaction amount as the weight value. Furthermore, once the values of RFM attributions are represented as a vector in three dimensions (3D) space, the Euclidian distance \( V \) can be calculated through:

\[
V = \sqrt{(W_R)^2 + (W_F)^2 + (W_M)^2}
\]  

(19)

where \( W_R \) is the SD of the R-value, \( W_F \) is the SD of the F-value, \( W_M \) is the SD of the M-value, and \( V \) is the customer value. A bigger (smaller) \( V \) represents higher (lower) customer value. Finally, the \( V \)-value is converted to percentile and entered into the CV Index. A \( V \)-value closer to one (zero) represents a higher (lower) CV Index.

5.2. Computational results and discussion

5.2.1. The evaluation and verification of the results of the algorithms
Once the customer transaction data from the practical case are analyzed and transformed through the RFM rule, it can be used as the test data for this empirical case and then to evaluate the performance among all algorithms. The parameter values for the proposed DCPIG algorithm are set according to Taguchi’s orthogonal arrays as shown in Table 4.

<table>
<thead>
<tr>
<th>Description of Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The total number of training (epochs)</td>
<td>500</td>
</tr>
</tbody>
</table>

PSO algorithm:
- Population size: 20
- Inertia weight (\( W \)): [1.2, 0.4]
- Learning factor (\( c_1, c_2 \)): \( c_1 = [0.35, 2.4], c_2 = [2.4, 0.35] \)
- The maximum velocity of each particle (\( V_{max} \)): 3

IG algorithm:
- Population size: 25
- Crossover rate: 0.6
- Mutation rate: 0.05
- Affinity rate (\( \alpha \)): 0.7
- Concentrate rate (\( \psi \)): 0.75
This empirical case adopts the Index value to evaluate the clustering results among all algorithms. Since the clustering results of the DCGA algorithm are unstable, as a result, the index value is not worthy of comparison. The averaged index values after 30-times trials among all algorithms applied to the customer data are listed in Table 5. The DCPIG algorithm is the best with the smallest SD among all algorithms, even when handling complex practical data and the number of clusters is unknown in advance.

The proposed DCPIG algorithm obtains the appropriate number of clusters using RFM attribute values from a customer transactional database. In the empirical case, the maximum numbers of clusters tested ranged from 15 to 26. The adequate number of clusters is determined along with the changing maximum number of clusters. Next, the Index value is calculated to evaluate the clustering performance of the DCPIG algorithm, as summarized in Table 6.

According to the numerical results in Table 6, the convergence situation is represented in Fig. 6. In Fig. 6, the average index value replaces the index value for the same Cluster #. Furthermore, we observed a sudden drop in Index value when the Cluster # is equal to 6. Therefore, the customer transaction data from the practical case is appropriately divided into 6 clusters. Afterward, this study views [0.0, 0.3], [0.4, 0.7],
Integration of PSO and IG Algorithm-Based Dynamic Clustering

Fig. 6. The Index values and the corresponding numbers of clusters for the proposed DCPIG algorithm.

Table 7. The clustering results obtained by the RFM model for the proposed DCPIG algorithm.

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Cluster # (%)</th>
<th>R</th>
<th>F</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14%</td>
<td>Hi</td>
<td>Hi</td>
<td>Hi</td>
</tr>
<tr>
<td>2</td>
<td>15%</td>
<td>Hi</td>
<td>Mi</td>
<td>Mi</td>
</tr>
<tr>
<td>3</td>
<td>19%</td>
<td>Lo</td>
<td>Hi</td>
<td>Hi</td>
</tr>
<tr>
<td>4</td>
<td>14%</td>
<td>Hi</td>
<td>Lo</td>
<td>Lo</td>
</tr>
<tr>
<td>5</td>
<td>18%</td>
<td>Lo</td>
<td>Lo</td>
<td>Hi</td>
</tr>
<tr>
<td>6</td>
<td>20%</td>
<td>Lo</td>
<td>Lo</td>
<td>Lo</td>
</tr>
</tbody>
</table>

and [0.8, 1.0] (i.e., RFM attribute values) as Lo (low), Mi (middle), and Hi (high) to evaluate the clustering results. The clustering analysis results of the practical case obtained through the RFM model for the DCPIG algorithm are presented in Table 7.

5.2.2. Empirical implication and analysis

According to the verified clustering results obtained through the proposed DCPIG algorithm, the customer transaction data in the practical case should be divided into 6 clusters. Therefore, the analysis and description of individual customer clusters is provided as follows.

Cluster 1: Advocates

This indicates the customers who proactively recommend and describe how good the brand is to others. They promote the brand voluntarily, which is the most powerful way to promote among all marketing approaches. This type of customer is highly loyal to the brand and is less sensitive to price. Besides, they are also more open to a wider range of products and services and use them more intensively. Accordingly, enterprises should consider hosting activities to intensify their loyalty, to learn customers’ needs and to offer them exclusive deals as well.
Cluster 2: Loyal customers
This represents the customers who consume regularly. Enterprises should implement loyalty programs to provide this type of customers with some rewards or something in return. In addition, enterprises should also dig into the requests and needs from this type of customer to increase their future loyalty and profit contribution.

Cluster 3: Repeating shoppers
This indicates customers who are willing to repeat purchase if they are satisfied by the quality of their first purchase. Enterprises should try to maintain the quality of products or services to build such confidence among customers. These customers should also be aware of coming products in advance. Enterprises should enhance the interaction between customers and strengthen their incentive to make repeat purchases.

Cluster 4: First-time shoppers
This represents the customers who make initial purchases for the merchant. The empirical results show that this type of customer is the majority in this case. If the company could use the web site to provide more detailed product information, real-time interaction, online security transactions, logistics and post-sales services, the customers’ switching costs would be higher. This would intensify their purchase experience in a positive way.

Cluster 5: Silent customers (high monetary value)
The reason customers leave is likely due to changing needs, bad communication with the merchant, low switching costs, etc. Enterprises should try hard to understand their reason of leaving, or provide marketing campaigns to promote the major products that customers need. Moreover, in this case, we found that customers in this cluster have relatively high monetary value. Therefore, if enterprises can allocate marketing resources to this type of customer, they can expect to see higher revenue.

Cluster 6: Silent customers (low monetary value)
This type of customer does not have any purchasing record, and possesses low shopping frequency and monetary value. As the profitability of this type of customer is far from the type in Cluster 5, enterprises could consider just investing certain retaining actions within a certain degree.

6. Conclusions
This study has proposed the DCPIG algorithm. This algorithm contains automatic clustering capability. It resolves the issue that the number of clusters needs to be pre-specified in most clustering algorithms. Besides, the Iris, Wine, Glass, and Vowel benchmark data sets were applied to evaluate the DCPIG algorithm. The computational results were then compared with the DCPG, DCPSO, and DCGA algorithms. Computational results indicate that the proposed DCPIG algorithm is able to combine the
Integration of PSO and IG Algorithm-Based Dynamic Clustering

advantages of both PSO and IG algorithms. It will not easily get stuck to the local optimum and is able to converge faster as well. In addition, the experimental results show that the DCPIG algorithm is able to automatically cluster data into an adequate number of clusters by examining data features. It can also achieve better and more stable clustering results compared to the other algorithms.

Furthermore, the proposed DCPIG algorithm was applied to a real-world problem considering the customer clustering problem of a cyber store. We use the RFM attributes from a database and group all customers into different clusters. As such, the enterprise is able to use the clustering results as a reference to understand the characteristics and preferences of different customer segments, and then to make the corresponding marketing strategies to raise its profitability.

In the future, the proposed DCPIG algorithm can be improved by further integrating it with other EAs to obtain better performance and further can also be applied to other industries.

Acknowledgments

This study was financially supported by the National Science Council of Taiwan Government under contract NSC99-2410-H-011-009-MY3. The support is appreciated. In addition, we also thank the case company for providing transaction data.

References

Integration of PSO and IG Algorithm-Based Dynamic Clustering

42. Z. Li, Y. Zhang and H. Z. Tan, An Efficient Artificial Immune Network with Elite-Learning, in *Proc. 3rd Int. Conf. on Natural Computation* (Haikou, China, 2007), Vol. 4, pp. 213–217.


