Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 19 (2009) 751–758
Progress in Natural Science
www.elsevier.com/locate/pnsc

Immunodominance and clonal selection inspired multiobjective clustering

Wenping Ma *, Licheng Jiao, Maoguo Gong

Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

Received 22 April 2008; received in revised form 21 July 2008; accepted 12 August 2008

Abstract

The biological immune system is a highly parallel and distributed adaptive system. The information processing abilities of the immune system provide important insights into the field of computation. Based on immunodominance in the biological immune system and the clonal selection mechanism, a novel data mining method, Immune Dominance Clonal Multiobjective Clustering algorithm (IDCMC), is presented. The algorithm divides an individual population into three sub-populations according to three different measurements, and adopts different evolution and selection strategies for each sub-population. The update of each sub-population, however, is not carried out in isolation. The periodic combination operation of the analysis of the three sub-populations represents considerable advantages in its global search ability. The clustering task is a multiobjective optimization problem, which is more robust with respect to the variety of cluster structures of different datasets than a single-objective clustering algorithm. In addition, the new algorithm can determine the number of clusters automatically, which should identify the most promising clustering solutions in the candidate set. The experimental results, using artificial datasets with different manifold structure and handwritten digit datasets, show that the IDCMC outperforms the PESA-II-based clustering method, the genetic algorithm-based clustering technique and the original *K*-Means algorithm in solving most of the problems tested.

Keywords: Artificial immune systems; Multiobjective optimization; Clustering; Unsupervised learning

1. Introduction

Unsupervised learning has attracted much interest from evolutionary computation (EC) researchers [1–5]. Recently, Handl and Knowles, for example, have proposed the multiobjective clustering technique MOCK [6], which shows good performance for solving data clustering problems unconventionally. They argue that the use of multiobjective optimization may provide a means to overcome some of the limitations of current algorithms. The simultaneous optimization of several complementary clustering objectives may lead to higher quality solutions and a more robust method of dealing with different data properties [7].

The biological immune system is a highly parallel and distributed adaptive system. The information processing abilities of the immune system provide important insights into the field of computation. This emerging field is sometimes referred to as immunological computation, immunocomputing, or artificial immune system (AIS). AIS which use immune system components and processes as the inspiration for constructing computational systems, have received a significant amount of interest from researchers and industrial sponsors in recent years. Applications of AIS include machine learning [8], fault diagnosis, computer security, scheduling, virus detection, and optimization [9–11].

* Corresponding author. Tel.: +86 29 88202661; fax: +86 29 88201023.
E-mail address: wpma@mail.xidian.edu.cn (W. Ma).

1002-0071/$ - see front matter © 2009 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2008.08.004

To maximize the performance of an AIS for unsupervised learning in real-world applications [12], one first has to carefully design an AIS (or choose an existing AIS), whose inductive bias is suitable for the target data and application domain. This paper focuses on the effectiveness of AIS in multiobjective clustering techniques. Multiobjective algorithms will always find solutions that are as good as, or better than, those of single-objective algorithms. In situations where the best solution corresponds to a trade-off between different objectives, a multiobjective algorithm is the only method that will be successful. Recently, an artificial immune system algorithm MISA [13] was proposed based on the clonal selection principle [14] to solve multiobjective optimization problems, and a vector artificial immune system (VAIS) based on the multimodal optimization algorithm opt-aiNet [16] was proposed by Freschi and Repetto [15].

In this paper, we introduce a novel multiobjective clustering algorithm, Immune Dominance Clonal Multiobjective Clustering algorithm (IDCMC). The experimental results, using six artificial datasets with different manifold structures and USPS handwritten digit datasets, show that the novel algorithm outperforms the MOCK algorithm based on PESA-II [17], the genetic algorithm-based clustering [18], and the *K*-Means algorithm [19] in solving most of the test problems.

2. Multiobjective clustering

We consider the clustering task as a multiobjective optimization problem, which seeks to minimize a vector of functions,

$$(P)\begin{cases}\min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))^T \\ \text{subject to} \quad \boldsymbol{x} \in \boldsymbol{X}\end{cases} \tag{1}$$

where $\boldsymbol{x}$ is a clustering of a given set of data $\boldsymbol{E}$, $\boldsymbol{X}$ is the set of feasible clusterings, and $f_i, i = 1, 2, \ldots, M$ is a set of M different criterion functions. Usually, no single best solution for this optimization task exists, but the framework of Pareto-optimality is embraced. It states that solution $\boldsymbol{x}_A \in \boldsymbol{X}$ dominates another solution $\boldsymbol{x}_B \in \boldsymbol{X}$ (written as $\boldsymbol{x}_A \prec \boldsymbol{x}_B$) if, and only if,

$$\forall i = 1, 2, \ldots, M, \quad f_i(\boldsymbol{x}_A) \le f_i(\boldsymbol{x}_B) \wedge \exists j = 1, 2, \ldots, M, \\ f_j(\boldsymbol{x}_A) < f_j(\boldsymbol{x}_B) \tag{2}$$

If no solution dominates $\boldsymbol{x}_A$, then $\boldsymbol{x}_A$ is a Pareto-optimal solution or nondominated solution. The set of all feasible Pareto-optimal solutions is referred to as a true Pareto-optimal set, while the corresponding image of objective vectors is called a true Pareto-optimal front.

The first multiobjective clustering algorithm VIENNA was based on PESA-II [17], and it uses two objectives. Handl and Knowles fine-tuned one of the objectives used in VIENNA, and also developed a method for determining the number of clusters automatically. These developments were incorporated in a new algorithm called MOCK [17]. MOCK consists of two main phases. In its initial clustering phase, MOCK uses a multiobjective evolutionary algorithm to optimize two complementary clustering objectives. The output of this first phase is a set of mutually nondominated clustering solutions, which correspond to different trade-offs between the two objectives, and also to different numbers of clusters. In the second model selection phase, MOCK analyzes the shape of the trade-off curve and compares it to the trade-offs obtained for an appropriate null model. Based on this analysis, the algorithm provides an estimate of the quality of all individual clustering solutions, and determines a set of potentially promising clustering solutions. Often, a single solution is clearly preferred and, in these cases, the number of clusters inherent to the dataset, k, is estimated implicitly.

In this paper, we select the two complementary objectives based on compactness and connectedness of clusters, respectively. The cluster compactness is simply computed as the overall summed distance between data items and their corresponding cluster center

$$Dev(\boldsymbol{x}) = \sum_{\boldsymbol{x}_k \in \boldsymbol{x}} \sum_{i \in \boldsymbol{x}_k} \delta(i, \mu_k) \tag{3}$$

where $\boldsymbol{x}$ is the set of all clusters, μ_k is the centroid of cluster $\boldsymbol{x}_k$, and $\delta(i, \mu_k)$ is the Euclidean distance between the ith data item of cluster $\boldsymbol{x}_k$ and μ_k. As an objective, overall deviation should be minimized. This criterion is similar to the well-known criterion of intra-cluster variance, which squares the distance value $\delta(i, \mu_k)$ and is more strongly biased towards spherically shaped clusters.

The cluster connectedness metric evaluates the degree to which neighboring data points have been placed in the same cluster. It is computed as

$$Conn(\boldsymbol{x}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} x_{i,n_{ij}} \right) \tag{4}$$

where $x_{r,s} = \begin{cases} \frac{1}{j}, \text{if } \bar{\exists}\, \boldsymbol{x}_k : r \in \boldsymbol{x}_k \wedge s \in \boldsymbol{x}_k \\ 0, \text{else} \end{cases}$ n_{ij} is the jth nearest neighbor of datum I, N is the size of the clustered dataset, and L is a parameter determining the number of neighbors that contribute to the connectivity measure. Eq. (4) gives more emphasis to the nearest neighbors, and its objective is to minimize connectivity. It therefore permits a finer distinction between the qualities of clustering solutions and allows for the identification of clusters of sizes significantly smaller than L.

3. Terms and operators

3.1. Terms

3.1.1. Antigen

In AIS, antigens (Ag) refer to problems and their constraints. For multiobjective optimization problems, the antigen is defined as the objective functions $\boldsymbol{F}(\boldsymbol{x})$ in Eq. (1).

3.1.2. Antibody and antibody population

In AIS, antibodies (Ab) represent candidates of the problem. The limited-length character string $\mathbf{a} = a_1 a_2 \ldots a_l$ is the antibody encoding of variable $\boldsymbol{x}$, denoted by $\mathbf{a} = e(\mathbf{x})$, and $\boldsymbol{x}$ is called the decoding of antibody $\boldsymbol{a}$, expressed as $\mathbf{x} = e^{-1}(\mathbf{a})$. Set $\boldsymbol{I}$ is called antibody space, namely $\mathbf{a} \in \mathbf{I}$. The antibody population $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \in \mathbf{I}^n$ is an n-dimensional group of antibody $\boldsymbol{a}$, namely,

$$\mathbf{I}^n = \{\mathbf{A} : \mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n), \mathbf{a}_k \in \mathbf{I}, 1 \leq k \leq n\} \tag{5}$$

where the positive integer n is the antibody population size.

3.1.3. Ab–Ag affinity

Ab–Ag affinity, i.e. the affinity between an antibody and an antigen, is the reflection of the total combination power between antigens and antibodies. In AIS, it generally indicates the values of objective functions or fitness measurement of the problem.

3.1.4. Ab–Ab affinity

Ab–Ab affinity, i.e. the affinity between two antibodies is the reflection of the total combined power between two antibodies. In this paper, we compute the Ab–Ab affinity as in Ref. [10]. Namely, if the coding of an antibody $\boldsymbol{a_i}$ is '1 1 0 0 0 0 1 0', and the coding of another antibody $\mathbf{a}_{di}$ is '1 1 0 1 0 1 1 0', then the number of genes matched between the two antibodies is 6. The matched gene strings whose lengths are greater than 2 are '110' and '10', and the corresponding lengths are 3 and 2, so that the Ab–Ab affinity between $\boldsymbol{a_i}$ and $\mathbf{a}_{di}$ is $6 + 3^2 + 2^2 = 19$. If the coding of antibodies is not a binary string, it should be converted to a binary string in advance.

3.1.5. Immune dominance

For a problem (P), the antibody $\boldsymbol{a_i}$ is an immune dominance antibody in a population $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$. If there is no antibody $a_j (j = 1, 2, \ldots, n \wedge j \neq i)$ in an antibody population, $\boldsymbol{A}$ satisfies (6):

$$\begin{aligned}&(\forall k \in \{1, 2, \ldots, p\} f_k(e^{-1}(\mathbf{a}_j)) \leq f_k(e^{-1}(\mathbf{a}_i))) \wedge \\ &(\exists l \in \{1, 2, \ldots, p\} f_l(e^{-1}(\mathbf{a}_j)) < f_l(e^{-1}(\mathbf{a}_i)))\end{aligned} \tag{6}$$

So the immune dominance antibodies are the Pareto-optimal individuals in the current population.

3.2. Operators

3.2.1. Clonal operation

In immunology, "clone" means asexual propagation, so that a group of identical cells can be descended from a single common ancestor, such as a bacterial colony whose members arise from a single original cell as the result of mitosis. In AIS, the clonal operation of the antibody population $\mathbf{A}(k)$ is defined as

$$Y(k) = T_{\mathrm{c}}^{\mathrm{C}}(A(k)) = T_{\mathrm{c}}^{\mathrm{C}}(a_1(k)) \left[T_{\mathrm{c}}^{\mathrm{C}}(a_2(k)), \ldots, T_{\mathrm{c}}^{\mathrm{C}}(a_{n_{\mathrm{b}}}(k))\right]^T \tag{7}$$

where $T_{\mathrm{c}}^{\mathrm{C}}(\mathbf{a}_{ci}(k)) = \mathbf{I}_{ci} \times \mathbf{a}_{ci}(k),\ i = 1, 2, \ldots, N$, $\boldsymbol{I_{ci}}$ is a q_{ci}-dimensional identity row vector. The process is called the q_{ci} clone of antibody a_i, namely $q_{ci}(k) = \hbar(n_{\mathrm{c}}, \boldsymbol{\Theta}_i)$, where $\boldsymbol{\Theta}_i$ stands for the affinity function of antibody a_i and other antibodies, and n_{c} is the clonal scale.

3.2.2. Immune differential degree

The immune differential degree denotes the relative distribution of an immune dominance antibody. Assuming that there are n_{d} immune dominance antibodies in the current population, f_{kl} is the value of the kth objective function of the lth antibody. The immune differential degree of the lth antibody a_l can be calculated as follows:

$$d_l^* = \min \left\{ \begin{array}{l} d_l(m) = \sqrt{\sum_{k=1}^{q} \left(\frac{\phi(f_{kl}) - \phi(f_{km})}{\phi(f_{kl})}\right)^2} \\ |l = 1, 2, \ldots, n_{\mathrm{d}};\ m = 1, 2, \ldots n_{\mathrm{d}} \wedge m \neq l \end{array} \right\} \tag{8}$$

where $\phi(\cdot)$ is an incremental function without the value of zero.

4. Description of the algorithm

Inspired by the concept of immunodominance from the biological immune system and the clonal selection mechanism, IDCMC is based on clonal selection with immune dominance and clone anergy for the multiobjective clustering problems, which can be implemented as follows:

Step 1. Give the termination generation $G_{\max}$, the size of the immune dominance antibody population n_{d}, the size of the generic antibody population n_{b}, the size of the dominance clonal antibody population n_{t}, and the clonal scale n_{c}. Set the mutation probability p_{m} and the recombination probability p_{c}. Generate the original antibody population $\mathbf{A}(0) = \{\mathbf{a}_1(0), \mathbf{a}_2(0), \ldots, \mathbf{a}_{n_{\mathrm{b}}}(0)\} \in \mathbf{I}^{n_{\mathrm{b}}}$, $k := 0$.

Step 2. Compute the Ab–Ag affinity of all the antibodies in $\boldsymbol{A}(k)$.

Step 3. According to the affinities, select all the immune dominance antibodies to constitute the population $\boldsymbol{DT}(k)$; if the number of antibodies in $\boldsymbol{DT}(k)$ is no larger than n_{d}, let the immune dominance antibody population $\boldsymbol{D}(k) = \boldsymbol{DT}(k)$, go to Step 6; otherwise go to Step 4.

Step 4. Compute the immune differential degrees of all the antibodies in the population $\boldsymbol{DT}(k)$.

Step 5. Sort all the antibodies in $\boldsymbol{DT}(k)$ in the descending order of their immune differential degrees, and select the first n_{d} antibodies as the current immune dominance antibody population $\boldsymbol{D}(k)$.

Step 6. If $k = G_{\max}$, export $\boldsymbol{D}(k)$ as the output of the algorithm, then stop; otherwise, replace the immune dominance antibody in $\boldsymbol{A}(k)$ by new antibodies generated randomly. Then mark the antibody population as $\boldsymbol{B}(k)$.

Step 7. Select an immune dominance antibody $\mathbf{a}_{di}$ randomly from $\boldsymbol{D}(k)$. Compute the Ab–Ab affinity between the antibodies in $\boldsymbol{B}(k)$ and the antibody $\mathbf{a}_{di}$.

Step 8. Sort all the antibodies in $\boldsymbol{B}(k)$ in the descending order of their Ab–Ab affinity, select the first n_t antibodies to constitute the dominance clonal antibody population $\boldsymbol{TC}(k)$, and the other antibodies to constitute the immune anergy antibody population $\boldsymbol{NR}(k)$.

Step 9. Implement the antibody clonal operation T_c^C at $\boldsymbol{TC}(k)$ according to Ab–Ab affinity and the clonal scale, and get the antibody population $\boldsymbol{CO}(k)$ after clonal operation.

Step 10. Implement the recombination operation at $\boldsymbol{CO}(k)$ with the probability p_c and get the antibody population $\boldsymbol{CO}'(k)$, $\mathbf{CO}'(k) = T_r^C(\mathbf{CO}(k))$; namely, for the antibody $\mathbf{y}_i(k)$ in $\boldsymbol{CO}(k)$, the following operation is implemented:\

$$\mathbf{y}_i'(k) = T_r^C(\mathbf{y}_i(k), \mathbf{a}_{di}(k)) = crossover(y_i(k), a_{di}(k)),$$
$$\mathbf{y}_i(k) \in \mathbf{CO}(k), \quad \mathbf{a}_{di}(k) \in \mathbf{D}(k) \tag{9}$$

where $crossover(y_i(k), a_{di}(k)), i = 1, 2, \ldots, n_c$ denotes selecting equiprobably one individual from the two offspring generated by a general crossover operator [20] on clone $\mathbf{y}_i(k)$ and an active antibody selected randomly from $D(k)$.

Step 11. Implement the mutation operation at $\mathbf{CO}'(k)$ with the probability p_m and get the antibody population $\boldsymbol{COT}(k)$, $\boldsymbol{COT}(k) = T_m^C(\mathbf{CO}'(k))$.

In this study, we use a static hypermutation operator [20] on the clone population after recombination, viz. the number of mutations is independent of the fitness values.

Step 12. Combine the populations $\boldsymbol{COT}(k)$, $\boldsymbol{D}(k)$ and $\boldsymbol{NR}(k)$ to form the antibody population $\boldsymbol{A}(k+1)$, $k := k + 1$, then go to Step 2.

IDCMC applies different update strategies to the three populations. At the beginning of IDCMC, the combination of $\boldsymbol{COT}(k)$, $\boldsymbol{D}(k)$ and $\boldsymbol{NR}(k)$ is propitious for increasing the global search ability. The update of the immune dominance antibody population remains the diversity of nondominated individuals. The update of the dominance clonal antibody population can select the dominant niche and assure the validity of the local search in the next generation. The existence of the immune anergy antibody population preserves the population diversity.

After running as normal until the maximum number of generations is reached, IDCMC returns a set of clustering solutions. These individual partitions correspond to different trade-offs between the two objectives and, in our case, also consist of different numbers of clusters. We apply an automated method for assessing the quality of individual clustering solutions proposed by Handl and Knowles [17]. This method can be used to identify one or more promising clustering solutions in the candidate set [21]. The selection of a single solution then automatically delivers an estimate of the number of clusters inherent in the dataset [22,23].

5. Analysis of the algorithm

5.1. Fitness assignment and population evolution

It can be seen that IDCMC uses some outstanding techniques on fitness assignment and population evolution, such as storing the nondominated solutions that were previously found externally and performing clustering to reduce the number of nondominated solutions stored without destroying the characteristics of the trade-off front. Other features of IDCMC can be characterized as follows:

(i) Its fitness values of current dominated individuals are assigned as the values of a custom distance measure, termed Ab–Ab affinity, between the dominated individuals and one of the nondominated individuals found so far.
(ii) All dominated individuals (antibodies) are divided into two kinds, dominance clonal antibodies and immune anergy antibodies, according to the values of Ab–Ab affinity.
(iii) Local search only applies to the dominance clonal antibodies. The immune anergy antibodies are redundant and have no function during local search, but they can become dominance clonal antibodies during the subsequent evolution.
(iv) A new immune operation, antibody clonal operation, is provided to enhance local search. Using the antibody clonal operation, IDCMC reproduces individuals and selects their improved matured progenies after local search. This allows single individuals to exploit their surrounding space effectively and the newcomers yield a broader exploration of the search space.

5.2. Population diversity

To approximate the Pareto-optimal set in a single run, multiobjective optimization evolutionary algorithms (MOEAs) have to perform a multimodal search where multiple, widely different solutions should be found. Therefore, maintaining a diverse population is crucial for the efficacy of an MOEA [24].

IDCMC adopts three strategies to preserve the population diversity. For the immune dominance antibody population, in the initial stage, we give a rough immune dominance antibody population, and make the immune dominance antibody population follow exactly along with the evolution search. This process is an interactive course, which is evolved through the concept of immune dominance.

By applying the strategies of clonal recombination and mutation to the dominance clonal antibody population, the algorithm can search locally or globally in many directions around one parent. Searching ability is very good, which makes the antibodies in the antibody population evolve quickly.

For the immune anergy antibody population, we do not adopt any operation, and the only purpose is to retain the diversity of the current antibody population and to maintain the forward progression of the evolution course.

5.3. Computational complexity

Analyzing IDCMC's computational complexity is revealing. In this section, we consider only the population size in

computational complexity. Assuming that the generic antibody population size is n_b, the immunodominant antibody population size is n_d, the dominance clonal antibody size is n_t, the immune anergy antibody population size is $n_b - n_t$, and the clone scale is n_c, then the time complexity of one iteration for the algorithm can be calculated as follows:

The time complexity for calculating Ab–Ag affinities is $O(n_b)$. The worst time complexity for the update of the immunodominant antibody population is $O((n_d + n_t \times n_c)^2)$. The time complexity for calculating the Ab–Ab affinities of all dominated antibodies is $O(n_b - n_t + n_t \times n_c)$. The worst time complexity for the updated dominance clonal antibody population and immune anergy antibody population is $O((n_d + n_b - n_t + n_t \times n_c) \log(n_d + n_b - n_t + n_t \times n_c))$. The time complexity for the clonal operation is $O(n_t \times n_c)$, and the time complexity for the recombination and mutation operation is $O(n_t \times n_c)$. Thus, the worst total time complexity is

$$O(n_b) + O((n_d + n_t \times n_c)^2) + O(n_b - n_t + n_t \times n_c) + O((n_d + n_b - n_t + n_t \times n_c) \log(n_d + n_b - n_t + n_t \times n_c)) + O(n_t \times n_c) + O(n_t \times n_c) \quad (10)$$

According to the operational rules of the symbol O, the worst time complexity of one generation for IDCMC can be simplified as follows:

$$O((n_d + n_t \times n_c)^2) + O((n_d + n_b - n_t + n_t \times n_c) \times \log(n_d + n_b - n_t + n_s \times n_c)) \quad (11)$$

If we denote the total size of all populations as N, namely, $N = n_d + n_b - n_t + n_t \times n_c$, then the computational complexity of IDCMC is

$$O((n_d + n_t \times n_c)^2) + O((n_d + n_b - n_t + n_t \times n_c) \times \log(n_d + n_b - n_t + n_s \times n_c)) < O(N^2) \quad (12)$$

6. Experimental study

6.1. Evaluation function of clustering performance

Clustering quality is evaluated using the adjusted Rand index (ARI) [25], which is a generalization of the Rand index [7]. The Rand indices take two partitionings as the input and count the number of pair-wise co-assignments of data items between the two partitionings. The adjusted Rand index additionally introduces a statistically induced normalization in order to yield values close to 0 for random partitions. Using a representation based on contingency tables [7], the adjusted Rand index is given as

$$R(U,V) = \frac{\sum_{lk}\binom{n_{lk}}{2} - \left[\sum_l \binom{n_{l\cdot}}{2} \sum_k \binom{n_{\cdot k}}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_l \binom{n_{l\cdot}}{2} + \sum_k \binom{n_{\cdot k}}{2}\right] - \left[\sum_l \binom{n_{l\cdot}}{2} \sum_k \binom{n_{\cdot k}}{2}\right] \Big/ \binom{n}{2}} \quad (13)$$

where n_{lk} denotes the number of data items that have been assigned to both cluster l and cluster k. The adjusted Rand index return value in the interval [1] is to be maximized. Let the true clustering be $\Delta^{true} = \{C_1^{true}, C_2^{true}, \ldots, C_{k_{true}}^{true}\}$ and the clustering produced be $\Delta = \{C_1, C_2, \ldots, C_k\}$. $\forall i \in [1, \ldots, k_{true}]$, $j \in [1, \ldots, k]$, *Confusion* (i,j) denotes the number of the same data points both in the true cluster C_i^{true} and in the cluster C_j produced. Then, the clustering error (CE) is defined as

$$CE(\Delta, \Delta^{true}) = \frac{1}{n} \sum_{i=1}^{k_{true}} \sum_{\substack{j=1 \\ i \neq j}}^{k} Confusion(i,j) \quad (14)$$

where n is the total number of data points. Note that there exists a renumbering problem. For example, cluster 1 in the true clustering might be assigned cluster 3 in the clustering produced. To counter this, the CE is computed for all possible renumbering of the clustering produced, and the minimum of all these is taken.

6.2. Simulation on artificial datasets

We present first the experimental results on six artificial datasets, named Long1, Square4, Size5, Sticks, Line-blobs and Three-circles in order to study a range of different interesting data properties. For illustrations of these demanding problems, see Handl [26]. In our experiments, we compare IDCMC with two classical clustering algorithms with proven performance across a wide range of datasets, the genetic algorithm-based clustering (GAC) [18] and the *K*-Means algorithm (KM) [19]. Moreover, to establish the usefulness of IDCMC, we also compare it with a multiobjective approach PESA-II based on the clustering method (PESC) [17]. In GAC and KM, the desired clusters number is set in advance.

For fair play, both IDCMC and PESC use the locus-based adjacency representation [17] and the initialization method based on minimum spanning trees [17]. The parameter settings used for IDCMC are as follows: the terminal generation $G_{max} = 100$, immune dominance antibody population size $n_d = 100$, generic antibody population size $n_b = 100$, dominance clonal antibody population size $n_t = 50$, maximum number of clusters 20, clonal scale $n_c = 300$, mutation probability $p_m = 1/N$, where N is the size of the dataset, and recombination probability $p_c = 0.7$. For PESC, the number of generations is 100, external population size is 100, internal population size is 100, maximum number of clusters is 20, mutation probability is $1/N$, and recombination probability is 0.7. For GAC, the number of generations is 100, population size is 50, recombination probability is 0.7, and mutation probability is 0.1. For KM, the maximum iterative number is set to 500, and the stop threshold 1e−10. Both algorithms are run 30 times for each of the candidate parameters. The average results are shown in Table 1.

The results of the ARI are presented graphically as boxplots [27] in Fig. 1. We can see clearly that on most datasets, the solutions generated by IDCMC are better than those of the other algorithms by a large margin. On Long1, Sticks, Line-blobs and Three-circles, its superiority is clear.

Table 1
Performance comparisons of IDCMC, PESC, GAC and KM on artificial datasets.

Problem	Clustering error			
	IDCMC	PESC	GAC	KM
Long1	**0**	0.020	0.445	0.486
Square4	0.086	0.090	**0.062**	0.073
Size5	0.015	**0.007**	0.023	0.024
Sticks	**0**	0.027	0.277	0.279
Line-blobs	**0**	0.013	0.263	0.256
Three-circles	**0.008**	0.014	0.569	0.545

Note. Results in bold are the best.

By comparison, GAC did best only on the Square4 dataset and PESC did best only on the Sizes5 dataset. KM and GAC only obtained desired clustering for the two spheroid datasets, i.e. Size5 and Square4, even though the desired clusters number is set in advance for GAC and KM. This is due to the complex structure of the other four datasets, which does not satisfy the super-sphere distribution. On the other hand, IDCMC and PESC can successfully recognize these complex clusters. When comparisons are made between IDCMC and PESC, the average value of ARI in solving the Long1, Sticks and Line-blobs obtained by IDCMC is 1, this shows that IDCMC can obtain the true clustering on the three problems in all the 30 runs. For the Long1, Square4, Sticks, Line-blobs and Three-circles, IDCMC did slightly better than PESC, but for Sizes5, IDCMC performed a little worse than PESC.

6.3. Simulation on handwritten digit datasets

We also performed experiments on a USPS handwritten digit dataset [28]. The USPS dataset contains 9298 16×16

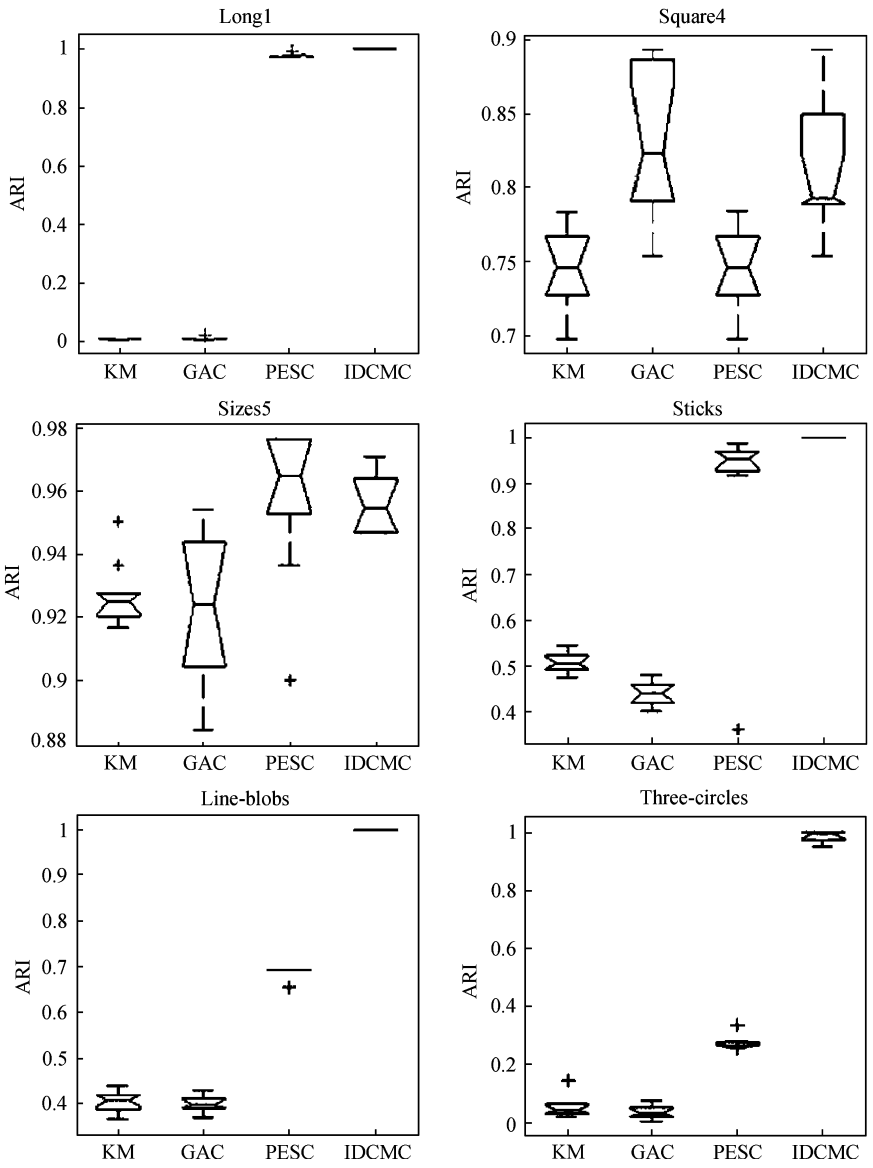

Fig. 1. Boxplots giving the distribution of ARI values achieved for 30 runs of each algorithm on artificial datasets.

Table 2
Performance comparison of IDCMC, PESC, GAC and KM on real-world datasets.

Problem	Clustering error			
	IDCMC	PESC	GAC	KM
{0,8}	**0.022**	0.039	0.105	0.191
{3,5,8}	**0.089**	0.096	0.275	0.352
{3,8,9}	**0.099**	0.120	0.207	0.386
{1,2,3,4}	**0.044**	0.056	0.204	0.304
{1,2,4,8}	**0.034**	0.058	0.207	0.307
{0,2,4,6,7}	**0.096**	0.135	0.168	0.202

Note. Results in bold are best.

gray images of handwritten digits (7291 for training and 2007 for testing). The test set is taken as the clustering data. We selected three hard datasets to recognize datasets {0,8}, {3,5,8}, {3,8,9}, and three relatively easy ones to recognize datasets {1,2,3,4}, {1,2,4,8}, {0,2,4,6,7}. The parameter settings were the same as those indicated in Section 6.2, and both algorithms were run 30 times for each of the digit datasets. The average values of clustering error achieved by IDCMC, PESC, GAC and KM on the six datasets are reported in Table 2. Fig. 2 visualizes the distribution of ARI values obtained by the different algorithms.

From Table 2 and Fig. 2, we can see clearly that on all datasets, whether hard or simple, the solutions generated by IDCMC have a better performance compared with those generated by PESC, GAC and KM. PESC is the second best, and KM is the worst. The latter result is because KM attempts to minimize the summed variance of points within each cluster from its centroid. Although such a method is very effective on certain sets of data, it is not robust enough to cope with variations in cluster shape, size, dimensionality and other characteristics.

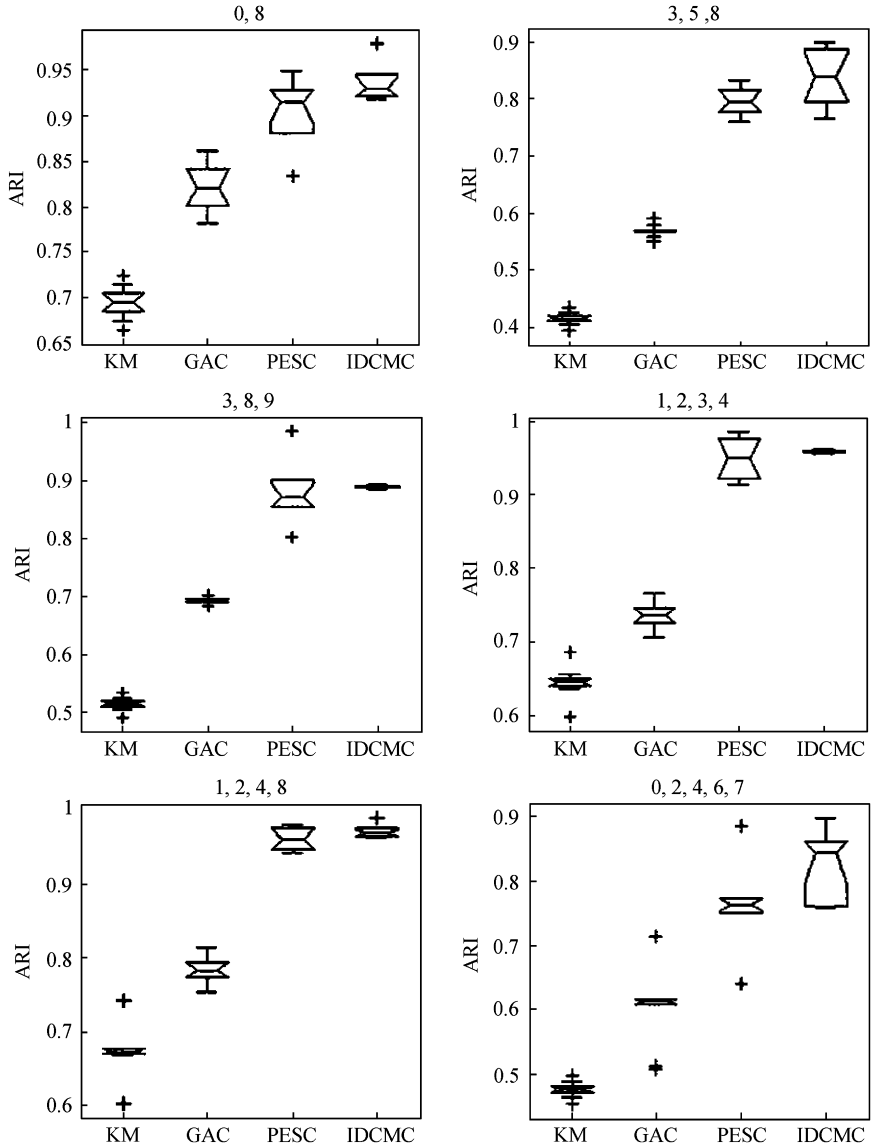

Fig. 2. Boxplots showing the distribution of ARI values achieved for 30 runs of each algorithm on the USPS handwritten digit datasets.

7. Conclusions

Based on the artificial immune system, a novel multiobjective clustering algorithm, immune dominance clonal multiobjective clustering algorithm is put forward. The clustering task is considered as a multiobjective optimization problem, which is more robust with regard to the variety of cluster structures of different datasets in comparison with the single-objective clustering algorithm. IDCMC is characterized by its unique fitness assignment strategy based on the Ab–Ab affinity and by its enhanced local research around nondominated individuals found so far using the clonal operation. In addition, the novel algorithm can determine the number of clusters automatically, which should identify the most promising clustering solutions in the candidate set. We demonstrated the clustering performance of IDCMC by an experimental study on six artificial datasets and on handwritten digit datasets. The results were compared with those of the PESA-II-based clustering method, the genetic algorithm-based clustering technique and the original *K*-Means algorithm. The experimental results of adjusted Rand index and clustering error indicate that IDCMC outperforms the other three clustering algorithms in solving most of the test problems.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60703107, 60703108), the National High Technology Research and Development Program (863 Program) of China (Grant Nos. 2006AA01Z107 and 2008AA12Z2475853), the National Basic Research Program (973 Program) of China (Grant No. 2006CB705700), the Program for New Century Excellent Talents in University, and the Program for Cheung Kong Scholars and Innovative Research Team in University (Grant No. IRT0645).

References

[1] Gong MG, Jiao LC, Ma WP, et al. Multiobjective optimization using an immuno-dominance and clonal selection inspired algorithm. Sci China (Ser F) 2008;51(8):1064–82.
[2] Freitas A, Timmis J. Revisiting the foundations of artificial immune systems for data mining. IEEE Trans Evol Comput 2007;11(4):521–40.
[3] Hall LO, Ozyurt IB, Bezdek JC, et al. Clustering with a genetically optimized approach. IEEE Trans Evol Comput 1999;3(2):103–12.
[4] Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. New York: John Wiley and Son Ltd.; 1990.
[5] Pan H, Zhu J, Han D. Genetic algorithms applied to multiclass clustering for gene expression data. Genomics Proteomics Bioinformatics 2003;1(4):279–87.
[6] Handl J, Knowles J. Evolutionary multiobjective clustering. In: Proceedings of the eighth international conference on parallel problem solving from nature. Berlin: Springer-Verlag; 2004. p. 1081–91.
[7] Rand W. Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 1971;66(336):846–50.
[8] Yoo J, Hajela P. Immune network simulations in multicriterion design. Struct Optimization 1999;18:85–94.
[9] Han JW, Kamber M. Data mining: concept and techniques. Vermont: Morgan Kaufman Publishers; 2000.
[10] Coello Coello CA, Cruz Cortes N. An approach to solve multiobjective optimization problems based on an artificial immune system. In: Proceedings of the first international conference on artificial immune systems, September vol. 9–11. Berlin: Springer; 2002. p. 212–21.
[11] Luh GC, Chueh CH, Liu WW, et al. MOIA: multi-objective immune algorithm. Eng Optimization 2003;35(2):143–64.
[12] De Castro LN, Timmis J. Artificial immune systems: a new computational intelligence approach. Berlin: Springer-Verlag; 2002.
[13] Coello Coello CA, Cortes NC. Solving multiobjective optimization problems using an artificial immune system. Genetic Programm Evolvable Mach 2005;6:163–90.
[14] Burnet FM. The clonal selection theory of acquired immunity. Cambridge: Cambridge University Press; 1959.
[15] Freschi F, Repetto M. Multi-objective optimization by a modified artificial immune system algorithm. Lecture Notes Comput Sci 2005;3627:248–61.
[16] deCastro LN, Timmis J. An artificial immune network for multimodal function optimization. In: Proceedings of the 2002 IEEE congress on evolutionary computation, 2002. p. 699–704.
[17] Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE Trans EC 2006;11(1):56–76.
[18] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. Pattern Recognit 2000;33(9):1455–65.
[19] Hartigan JA, Wong MA. A K-Means clustering algorithm. Appl Stat 1979;28:100–8.
[20] Cutello V, Nicosia G, Pavone M. Exploring the capability of immune algorithms: a characterization of hypermutation operators. Lecture Notes Comput Sci 2004;3239:263–76.
[21] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Stanford University, vol. 63, No. 2; 2000. p. 411–23.
[22] Handl J, Knowles J, Kell DB. Computational cluster validation in post-genomic data analysis. Bioinformatics 2005;21(5):3201–12.
[23] Sarle WS. Cubic clustering criterion. SAS Technical Report A-108, Cary, NC: SAS Institute Inc., Technical Report; 1983.
[24] Zitzler E. Evolutionary algorithms for multiobjective optimization: methods and applications. Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich; 1999.
[25] Hubert A. Comparing partitions. J Classif 1985;2:193–8.
[26] Handl J. Supporting material. Available from: http://dbk.ch.umist.ac.uk/handl/vienna/ [2008-04-15].
[27] Weisstein EW. Box-and-whisker plot. http://mathworld.wolfram.com/Box-and-WhiskerPlot.html [2008-04-15].
[28] LeCun Y, Boser B, Denker JS, et al. Back propagation applied to handwritten zip code recognition. Neural Comput 1989;1(4):541–51.