An unsupervised learning algorithm for membrane computing

Hong Peng [a], Jun Wang [b], Mario J. Pérez-Jiménez [c], Agustín Riscos-Núñez [c]

[a] Center for Radio Administration and Technology Development, Xihua University, Chengdu, Sichuan 610039, China
[b] School of Electrical and Information Engineering, Xihua University, Chengdu, Sichuan 610039, China
[c] Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, University of Seville, Sevilla 41012, Spain

ABSTRACT

This paper focuses on the unsupervised learning problem within membrane computing, and proposes an innovative solution inspired by membrane computing techniques, the fuzzy membrane clustering algorithm. An evolution–communication P system with nested membrane structure is the core component of the algorithm. The feasible cluster centers are represented by means of objects, and three types of membranes are considered: evolution, local store, and global store. Based on the designed membrane structure and the inherent communication mechanism, a modified differential evolution mechanism is developed to evolve the objects in the system. Under the control of the evolution–communication mechanism of the P system, the proposed fuzzy clustering algorithm achieves good fuzzy partitioning for a data set. The proposed fuzzy clustering algorithm is compared to three recently-developed and two classical clustering algorithms for five artificial and five real-life data sets.

1. Introduction

The learning problem aims at finding hidden patterns in data or estimating the unknown model parameters from the given data. From the viewpoint of machine learning, learning methods can be categorized into three main variants: supervised learning, unsupervised learning and reinforcement learning. Unsupervised learning, also known as clustering, is the process of finding natural groupings from unlabeled data [8], that is, finding k clusters from a set of n data points according to some similarity measure such that patterns within the same cluster are more similar than those from different clusters. Over the past years, a large number of unsupervised learning methods have been introduced [13,47]. These methods can be fall into three categories: hierarchical, partitioned, and overlapping methods. Hierarchical methods can be either agglomerative, which begin with each element as a separate cluster and merge them into larger clusters, or divisive, which begin with the whole set and successively divide it into smaller clusters. Partitioned methods attempt to directly decompose the data set into several disjoint clusters without the hierarchical structure, while overlapping methods search soft or fuzzy partitioning by relaxing the mutually disjoint constraint.

K-means is one of most popular unsupervised learning methods due to its simplicity and effectiveness, and has been used in a wide variety of areas such as pattern recognition, data mining, and bioinformatics. However, k-means has several drawbacks: it is problematic to avoid local minima, it is sensitive to initial cluster centers, and it takes significant time to search the global optimal solution when the number of data points is large. To overcome these, evolutionary clustering algorithms have been considered in recent years based on genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), differential evolution (DE), artificial bee colony (ABC), and black hole (BH) algorithm.

GA-based clustering methods were first reported in the literature [5,17], and have two approaches to express the solution of a clustering problem: point-based schemes [18,23] and center-based schemes [2]. Point-based schemes can suffer from large search space and high computing cost when data points proliferate, so center-based schemes are used by most evolutionary clustering methods developed in recent years. Kao et al. [14] proposed a clustering method that uses PSO to optimize a set of cluster centers. Shelokar et al. [40] introduced an ACO-based method to find appropriate cluster centers. Das et al. [6] used DE in image pixel clustering, while Karaboga et al. [15] presented an ABC-based clustering method. Hatamlou [11] proposed a new optimization approach based on black hole for data clustering. In addition, several evolutionary clustering algorithms with hybrid mechanisms have been developed in recent years [16,24,25]. Fuzzy c-means (FCM) is a well-known overlapping clustering technique that uses the principles of fuzzy sets to evolve a partitioning matrix $U(X)$ [3]. However, FCM still has the limitation of finding suboptimal solutions. To overcome this, Maulik et al. [19] proposed a GA-based fuzzy clustering technique, Fuzzy-VGA, which automatically evolves the appropriate fuzzy partitioning for a data set by optimizing the well-known XB-index. Saha et al. [36] developed a fuzzy variable string length genetic point symmetry (Fuzzy-VGAPS) based clustering technique, where membership values of points to different clusters are computed based on a point symmetry-based distance rather than the Euclidean distance. Maulik et al. [20] presented a modified DE based fuzzy clustering (MoDEFC) algorithm and applied it to deal with pixel classification of remote sensing imagery. Sanchez et al. [39] proposed a fuzzy granular gravitational clustering algorithm for multivariate data. Saha et al. [37] developed a multiobjective modified differential evolution based fuzzy clustering algorithm.

Membrane computing was initiated by Gh. Păun [28], as a new branch of natural computing, aiming to abstract computing models from the structure and functioning of living cells, as well as from the cooperation of cells in tissues, organs, and cell populations [10,32]. Following this inspiration, classes of distributed parallel computing models have been defined, usually known as P systems or membrane systems. P systems have several interesting features: non-determinism, programmability, extensibility, readability, they are easy to communicate, etc., and many variants have been proposed [9,12,27,33,45,48,51]. Most P systems variants have proved to be powerful (in the sense of Turing completeness) and effective (since they have successfully solved a large number of NP-hard problems in a linear or polynomial time [31]). In recent years, the potentiality and characteristics of membrane computing have attracted much attention in relation to real-life applications, such as membrane algorithms for solving optimization problems [26,34,49,50,52], and fuzzy spiking neural P systems for dealing with knowledge representation and fault diagnosis [35,42,44].

Learning capability is an important and useful characteristic of natural computing methods, such as neural computing and evolutionary computing [21,22,38]. Unfortunately, existing P systems variants generally lack learning capability, due to their orientation towards a formal language framework. Therefore, Gh. Păun has listed providing learning capability to P systems as an interesting open problem [29,30].

The main motivation behind the work presented in this paper is to focus on the problem of unsupervised learning in membrane computing, and to propose a novel clustering method in this framework to solve fuzzy clustering problems: the fuzzy membrane clustering algorithm. An evolution–communication P system with a nested membrane structure is considered as the computing framework, and both its evolution–communication mechanism and the differential evolution mechanism of DE are integrated into a new P system based learning mechanism. The scientific contribution of this paper has two aspects: (i) this is the first attempt to use P systems for solving fuzzy clustering problems and (ii) a novel clustering method for data mining and/or machine learning is developed.

Section 2 describes the fuzzy clustering problem and several clustering validity indexes. In Section 3, we briefly review evolution–communication P systems and classical DE algorithms. Section 4 describes the proposed evolution–communication P system, and details the proposed fuzzy membrane clustering algorithm. Experiments and results are provided in Section 5. Finally, Section 6 draws the conclusions.

2. Problem statement

2.1. Fuzzy clustering problem

Data clustering can be seen as the task of distributing (partitioning) n data points into several groups according to some similarity measure. Let us consider $X = \{X_1, X_2, \ldots, X_n\} \subseteq R^{n \times d}$, a set of n unlabeled data points in a d-dimensional Euclidean space, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, that is, x_{ij} is the j-th real-value feature of the i-th data point. The goal of a fuzzy clustering algorithm is to find a fuzzy partition of the data set, X, into k clusters, $C_1, C_2, \ldots, C_K$, in such a way that the similarity of data points in the same cluster is very high, and at the same time the similarity of data points in different clusters is very low. In the context of fuzzy clustering, a data point may belong to each of the clusters with a certain fuzzy membership degree, which can be captured by a fuzzy partitioning matrix, $U_{K \times n}$, where each u_{ij} is the fuzzy membership degree of X_j to C_i. The elements of such matrix should satisfy the following properties:

$$\begin{cases} 0 < \sum_{j=1}^{n} u_{ij} < n & \text{for } i = 1, 2, \ldots, K \\ \sum_{i=1}^{K} u_{ij} = 1 & \text{for } j = 1, 2, \ldots, n \\ \sum_{i=1}^{K} \sum_{j=1}^{n} u_{ij} = n \end{cases} \qquad (1)$$

Thus, for a given data set, solving a fuzzy clustering problem is equivalent to finding the optimum fuzzy partition. Since center-based representation is used to express a clustering solution, fuzzy clustering can be considered as finding the optimal cluster centers. Therefore, the fuzzy clustering problem can be posed as the following optimization problem:

$$\arg\min_{z_1,z_2,\ldots,z_K}[J(z_1,z_2,\ldots,z_K)=f(z_1,z_2,\ldots,z_K)], \tag{2}$$

where $z_1, z_2, \ldots, z_K$ are K parameters to be optimized, which denote a candidate set of cluster centers; and $f(z_1,z_2,\ldots,z_K)$ is its objective function.

2.2. Clustering validity index

To evaluate the performance of clustering algorithm, it is necessary to use a well-defined statistical–mathematical function, the cluster validity index, to measure the quality of the obtained solution. We use four well-known clustering validity indices, described below, in our experiments.

2.2.1. Xie–Beni index

The Xie–Beni (XB) index is defined as a function of the ratio of total variation, σ, to minimum separation, sep, of the clusters [46],

$$\sigma(U,Z;X) = \sum_{i=1}^{K}\sum_{j=1}^{n}u_{i,j}^2\|X_j - z_i\|^2,$$

and

$$sep(Z) = \min_{i\neq j}\|z_i - z_j\|^2,$$

where $\|\cdot\|$ is the Euclidean norm, and

$$u_{i,j} = \frac{\left(\frac{1}{\|X_j - z_i\|}\right)}{\sum_{i=1}^{K}\left(\frac{1}{\|X_j - z_i\|}\right)}.$$

Thus, the XB index is

$$XB(U,Z;X) = \frac{\sigma(U,Z;X)}{n \times sep(Z)}. \tag{3}$$

Note that when a partitioning is compact and good, σ should be low while sep should be high, thereby yielding lower values of XB index. In this work, the XB index is used as an objective function of the fuzzy clustering problem, and it is therefore minimized to achieve a good partitioning.

2.2.2. PBMF index

The PBMF index is defined as

$$PBMF(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K\right)^2, \tag{4}$$

where K is the number of clusters,

$$E_K = \sum_{i=1}^{K}\sum_{j=1}^{n}u_{i,j}\|X_j - z_i\|,$$

and

$$D_K = \max_{i\neq j}\|z_i - z_j\|.$$

In Eq. (4), E_1 usually appears as a constant for a given data set. Generally, larger PBMF implies a better clustering solution.

2.2.3. FCM measure (J_m)

The FCM measure (J_m), corresponds to the total variation of data points with respect to cluster centers, taking into consideration the membership given by the fuzzy partitioning matrix, and is defined as the total mean square error (MSE),

$$J_m(C_1,C2,\ldots,C_K) = \sum_{i=1}^{K}\sum_{j=1}^{n}u_{i,j}^2\|X_j - z_i\|^2 \tag{5}$$

Generally, smaller J_m indicates a better clustering solution.

2.2.4. F-Measure

Let m_{ij} be the number of points that belong to both cluster i and cluster j, m_i the total number of points in cluster i. The F-Measure $(F(i,j))$ of cluster i with respect to cluster j is

$$F(i,j) = \frac{(2 \times precision(i,j) \times recall(i,j))}{(precision(i,j) + recall(i,j))}, \tag{6}$$

where $precision(i,j) = p_{ij} = m_{ij}/m_i$ expresses the precision of cluster i with respect to cluster j, and $recall(i,j) = m_{ij}/m_j$ denotes the recall of cluster i with respect to cluster j. Thus, the overall F-Measure of the whole partitioning is

$$F = \sum_j \frac{m_j}{m} \max_i F(i,j) \tag{7}$$

For F-Measure, the optimum score is 1, with higher scores implying a better clustering solution.

3. Methods

3.1. Evolution–communication P systems

Researchers have proposed a variety of P systems, drawing concepts from biology, mathematics, computer science, etc. Cavaliere [4] proposed an evolution–communication P system, which contains two types of rules: evolution rules (classical evolution rules without communication targets) and symport/antiport rules (for communication). The motivation of the variant was to split the P system in two phases: the evolution of the objects (application of evolution rules) and the communication between the regions of the system (application of the symport/antiport rules). More details of this variant can be found in [1,4].

An evolution–communication P system of degree $q \geqslant 1$ is defined as

$$\Pi = (O, \mu, w_1, w_2, \ldots, w_q, R_1, R_2, \ldots, R_q, R'_1, R'_2, \ldots, R'_q, i_o),$$

where

(1) O is the alphabet of objects.
(2) μ is a membrane structure with q nested membranes (and hence q regions), labeled by $1, 2, \ldots, q$, respectively.
(3) $w_1, w_2, \ldots, w_q$ are initial multisets of objects over O placed in the regions $1, 2, \ldots, q$ of μ.
(4) $R_1, R_2, \ldots, R_q$ are finite sets of evolution rules of the q regions respectively. An evolution rule is of the form $u \rightarrow v$, where $u, v \in O^*$.
(5) $R'_1, R'_2, \ldots, R'_q$ are finite sets of communication rules of the q membranes. The communication rules are of two forms: $u \rightarrow [u]_i$, and $[u]_i \rightarrow u$, where $u \in O^*$.
(6) $i_o \in \{0, 1, 2, \ldots, q\}$ is the output region. If $i_o = 0$, then the environment is the output region, otherwise the region of membrane i_o is the output region.

Symport/antiport rules are commonly used in the tissue-like framework, and this evolution–communication P system is the first case where symport rules are used in a cell-like variant (i.e., over a nested membrane structure). Each region has several evolution rules, of the form $u \rightarrow v, u, v \in O^*$, and the application of such rule means that u is changed to v (multiset u is "consumed" while multiset v is "generated" instead). Each membrane has also some communication rules, which are used to exchange the objects between the inner region and the outer region of the membrane. Two types of symport rules are considered: $u_1 u_2 \cdots u_s \rightarrow [u_1 u_2 \cdots u_s]_i$, and $[u_1 u_2 \cdots u_s]_i \rightarrow u_1 u_2 \cdots u_s$, where $u_1 u_2, \ldots, u_s \in O^*$. Execution of first rule means that objects $u_1, u_2, \ldots, u_s$ are transmitted into the inner region from the outer region, while second rule indicates that objects $u_1, u_2, \ldots, u_s$ are exchanged from the inner region to the outer region.

As usual in membrane computing, each membrane operates as a parallel computing unit, with a universal clock considered for synchronization. A computation is a sequence of steps which start with the membranes $1, \ldots, q$ containing the multisets $w_1, \ldots, w_q$. In each step, one or more rules are applied to the current multisets of objects, in a maximally parallel manner. A computation is successful if and only if it halts. When it halts, it produces a result in the output region.

3.2. Classical DE algorithm

DE is arguably one of the most powerful population-based stochastic optimization algorithms [41]. It searches a global optimal solution using three evolution operations (mutation, crossover and selection) [7].

Let Z_i be a chromosome/genome in the considered population. A donor vector Y_i is created to change original vector Z. There are five frequently used schemes that can be used to create this donor vector:

(1) DE/rand/1: $Y_i = Z_{r_1} + F \cdot (Z_{r_2} - Z_{r_3})$,

(2) DE/best/1: $Y_i = Z_{best} + F \cdot (Z_{r_1} - Z_{r_2})$,

(3) DE/current-to-best/1": $Y_i = Z_i + F \cdot (Z_{best} - Z_i) + F \cdot (Z_{r_1} - Z_{r_2})$,

(4) DE/best/2: $Y_i = Z_{best} + F \cdot (Z_{r_1} - Z_{r_2}) + F \cdot (Z_{r_3} - Z_{r_4})$,

(5) DE/rand/2: $Y_i = Z_{r_1} + F \cdot (Z_{r_2} - Z_{r_3}) + F \cdot (Z_{r_4} - Z_{r_5})$.

The indices, r_1, r_2, r_3, r_4, and r_5, are mutually exclusive integers randomly chosen, and all of them different from the base index, i. The scaling factor, F, is a positive control parameter for scaling the difference vectors. Z_{best} is the best individual vector with the best fitness in the population.

To increase the potential diversity of the population, a crossover operation is applied. The DE-family uses two crossover schemes, exponential and binormal. We briefly describe the binormal crossover, which we employ in the improved DE mechanism. Binormal crossover is performed on each of the D variables whenever a randomly generated number between 0 and 1 is less than or equal to the C_r value,

$$r_{ij} = \begin{cases} y_{ij}, & \text{if } \text{rand}_{ij}[0,1] \leqslant C_r \text{ or } j = j_{rand}, \\ z_{ij}, & \text{otherwise}, \end{cases} \tag{8}$$

where $\text{rand}_{ij}[0,1]$ is a uniformly distributed random number, which is called anew for each j-th component of the i-th parameter vector. $j_{rand} \in \{1, \ldots, D\}$ is a randomly chosen index, which ensures that R_i gets at least one component from Y_i.

The selection operation is used to determine which of the target vector or trial vector will survive in the next generation. If the trial vector yields a better value of the objective function, it will replace its target vector in the next generation; otherwise the parent is retained in the population:

$$Z_i' = \begin{cases} R_i, & \text{if } f(R_i) \leqslant f(Z_i), \\ Z_i, & \text{otherwise}, \end{cases} \tag{9}$$

where $f(\cdot)$ is the function to be minimized.

4. Proposed fuzzy membrane clustering algorithm

4.1. The evolution–communication P system

The proposed clustering algorithm is a fuzzy clustering algorithm inspired by membrane computing. An evolution–communication P system with $(2q + 1)$ membranes is considered as its computing core, which has a nested structure of three layers: an outermost (skin) membrane contains q membranes and each of them has an elementary (innermost) membrane, shown in Fig. 1. Membranes $1, \ldots, q$ are evolution membranes, since they are the only ones having evolution rules. Elementary membranes $1', 2', \ldots, q'$ are local store membranes, as each will store the best object found in the corresponding surrounding evolution membrane. The skin or global store membrane is considered to store the best object found in the entire system (i.e., all evolution membranes). Evolution membranes have evolution rules, whereas skin and local store membranes contain no evolution rules. Fig. 1 shows communication channels of the system that enable evolution membranes to exchange and share their objects with local store membranes and the skin membrane, respectively.

4.1.1. Objects

The evolution–communication P system has been designed to optimize a set of cluster centers for a fuzzy clustering problem, so each of its objects will express a feasible solution (a set of cluster centers). Let $X = \{X_1, X_2, \ldots, X_n\} \subset R^{n \times d}$ be a data set divided into K clusters, $C_1, C_2, \ldots, C_K$. We denote the centers of these K clusters as $z_1, z_2, \ldots, z_K$, where $z_i \in R^d$, for $i = 1, \ldots, K$. Fig. 2 shows the corresponding relationship between a solution (i.e., a set of cluster centers) and an object in the membranes. Thus, each object is designed as a $(K \times d)$-dimensional vector:

$$O = (z_{11}, z_{12}, \ldots, z_{1d}, \ldots, z_{i1}, z_{i2}, \ldots, z_{id}, \ldots, z_{K1}, z_{K2}, \ldots, z_{Kd}),$$

where $(z_{i1}, z_{i2}, \ldots, z_{id})$ corresponds to the i-th cluster center z_i, $i = 1, \ldots, K$.
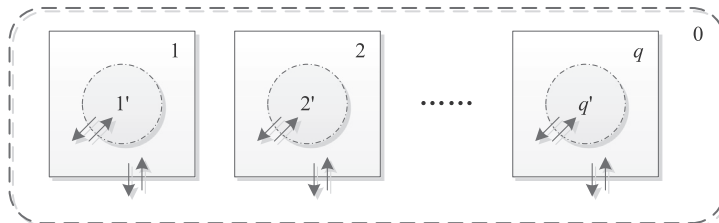

Fig. 1. The membrane structure of the evolution–communication P system and its communication channels.
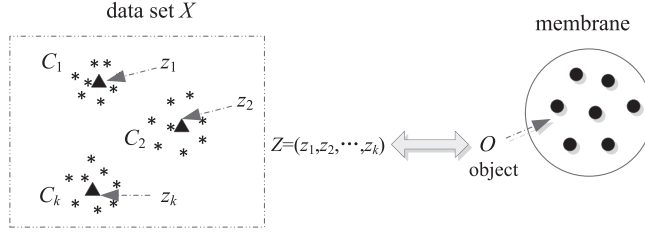
Fig. 2. The representation of objects in membranes.

Each evolution membrane contains a number of objects, each of which expresses a set of (feasible) cluster centers. For the sake of simplicity, let us assume that each evolution membrane has m objects. Each local store membrane, i', has only one object, the local best object, O_{lbest}^i, while the global store membrane also contains a single object, the global best object O_{gbest}.

Before the computation begins, the P system generates m initial objects for each evolution membrane. For each of such initial objects, $(K \times d)$ random real numbers are generated, which are required to satisfy the following constraints:

$$A_1 \leqslant z_{i1} \leqslant B_1, \ A_2 \leqslant z_{i2} \leqslant B_2, \ \ldots, \ A_j \leqslant z_{ij} \leqslant B_j, \ \ldots, \ A_d \leqslant z_{id} \leqslant B_d,$$

where A_j and B_j are lower and upper bounds on the j-th component of data points in X, respectively.

4.1.2. Evolution rules

The evolution of objects in the system is achieved only within the q evolution membranes. In this work, the differential evolution mechanism considered relies on evolution rules, which contain three evolution operators: mutation, crossover and selection. A modified version has been developed, based on the membrane structure and communication mechanism of evolution–communication P systems. The modification is inspired from the observation that each evolution membrane has two best objects which will take part in the evolution of objects, being the local best object, O_{lbest}^i, communicated from the corresponding local store membrane, and the global best object, O_{gbest}, communicated from the skin membrane. This modification is realized only on the mutation operator, while original crossover and selection operators are retained. Therefore, we only describe the modified mutation operator, which can be regarded as an extended version of the DE/best/1 scheme.

Let O_j^i be j-th object in evolution membrane i. Then the mutation object can be generated by

$$Y_j^i = \begin{cases} O_{gbest} + \alpha \cdot (O_{lbest}^i - O_j^i) + \alpha \cdot (O_{r_1}^i - O_{r_2}^i), & \text{if rand}(0,1) < \alpha \\ O_j^i + F \cdot (O_{r_3}^i - O_{r_4}^i), & \text{otherwise} \end{cases}, \tag{10}$$

where O_{lbest}^i is the local best object; O_{gbest} is the global best object; and $O_{r_1}^i$, $O_{r_2}^i$, $O_{r_3}^i$, and $O_{r_4}^i$ are four different objects chosen randomly from the evolution membrane, i. The scaling factors are

$$\alpha = \frac{1}{1 + \exp\left[-\frac{1}{t_{max}}\right]},$$
$$F = 0.5 \times (1 + \text{rand}(0,1)),$$

where t_{max} is the maximum execution step number (maximum number of iterations).

4.1.3. Communication rules

As discussed above, communication of objects is realized by communication rules, which enable communication channels connecting evolution membranes and local store membranes. Local store membranes have two types of communication rules, which exchange objects to and from evolution membranes:

(i) Rule $O_{best}^i \rightarrow [O_{best}^i]_{i'}$, where O_{best}^i is the best object in evolution membrane i, and $i = 1, \ldots, q$.

The rule communicates the best object, O_{best}^i, in evolution membrane i into local store membrane i' and updates its local best object, O_{lbest}^i. The updating strategy is

$$O_{lbest}^i = \begin{cases} O_{best}^i, & \text{if } f(O_{best}^i) < f(O_{lbest}^i) \\ O_{lbest}^i, & \text{otherwise} \end{cases}, \tag{11}$$

where $f(O)$ denotes the fitness value of object O.

(ii) Rule $[O_{lbest}^i]_{i'} \rightarrow O_{lbest}^i$, where O_{lbest}^i is the local best object in local store membrane i', and $i = 1, \ldots, q$.

The rule indicates that local best object in local store membrane i' is transmitted into evolution membrane i. The transmitted object will get engaged in the evolution of objects in the next computing step.

Evolution membranes also have two types of communication rules, which exchange objects to and from the skin membrane.

(i) Rule $[O_{best}^i]_i \rightarrow O_{best}^i$, where O_{best}^i is the best object in evolution membrane i, for $i = 1, \ldots, q$.

The rule transmits its best object, O_{best}^i, into the skin membrane and updates the global best object, O_{gbest}. The updating strategy is

$$O_{gbest} = \begin{cases} O_{best}^i, & \text{if } f(O_{best}^i) < f(O_{gbest}) \\ O_{gbest}, & \text{otherwise} \end{cases}, \tag{12}$$

(ii) Rule $O_{gbest} \rightarrow [O_{gbest}]_i$, where O_{gbest} is the global best object in the skin membrane, and $i = 1, \ldots, q$.

The rule indicates that the global best object, O_{gbest}, in the skin membrane is communicated into evolution membrane i. The communicated global best object will get engaged in the evolution of objects in the next computing step.

Note that the local best object and global best object participate in the evolution of objects during the computation, and these objects come from different sources. This brings the benefit of enhancing the diversity of objects in the system and avoiding the object's degradation.

4.1.4. Halting and output

For simplicity, the designed evolution–communication P system uses the halt condition of maximum number of execution steps. Upon halting, the global best object stored in the skin membrane is regarded as the final computing result, i.e., the best cluster centers found.

4.2. Fuzzy membrane clustering algorithm

The proposed evolution–communication P system is a core component of the proposed membrane clustering algorithm. The evolution–communication P system can find the optimal cluster centers for a data set based on its evolution–communication mechanism.

Given an instance of a fuzzy clustering problem consisting of n data points to be divided into K clusters, the evolution–communication P system uses objects to express sets of feasible cluster centers, and derives the optimal cluster centers by the evolution and communication of objects. The membrane clustering algorithm first generates m initial objects for each evolution membrane, and then executes the evolution–communication P system. Under the control of the modified differential evolution mechanism and communication mechanism, the evolution–communication P system evolves the objects in the system, updating local best objects $O_{lbest}^i (i = 1, \ldots, q)$ and the global best object O_{gbest} synchronously until the system halts. When the system halts, the global best object in the skin membrane is the best approximation to optimal cluster centers. We re-compute the partition matrix $U(X)$ according to the cluster centers found, and assign the n data points to the K clusters based on the principle of maximum membership.

The proposed membrane clustering algorithm is summarized in Table 1, based on the evolution–communication P system with a nested structure of three layers.

5. Experimental results

The proposed fuzzy membrane clustering algorithm (Fuzzy-MC) was compared with the improved differential evolution-based (Improved-DE), original differential evolution-based (Original-DE), genetic algorithm-based fuzzy clustering (GA) methods, classical fuzzy c-means (FCM) algorithm, and hierarchical average linkage (AL) clustering. Experimental results were calculated for five artificial and five real-life data sets, described below.

5.1. Artificial data sets

- Data 1: A highly overlapping two-dimensional data set with five clusters, and 250 points (Fig. 3(a)). Each cluster is circular.
- Data 2: A highly overlapping two-dimensional data set with nine clusters and 900 points (Fig. 3(b)). Each cluster is circular, and centers are on an approximate regular grid.
- Data 3: A two-dimensional data set with four clusters and 600 points (Fig. 3(c)). Each cluster is circular, with centers approximately collinear and approximately evenly distributed.

Table 1
Fuzzy membrane clustering algorithm: a fuzzy clustering algorithm based on evolution–communication P system.

Input: Data set, X; the number of clusters, K; the number of evolution membranes, q; the number of objects in each evolution membrane, m; maximum execution step number, t_{max}; and crossover probability, C_r.

Output: The optimal cluster centers, O_{gbest}, in the skin membrane.

Begin
  Step1: /∗Initialization∗/
  for $i = 1$ to q
      for $j = 1$ to m
         Generate j-th initial object for evolution membrane i, O_j^i;
         Calculate partition matrix, U_j^i;
         Calculate the XB index of object, O_j^i;
      end for
      Update local best object, O_{lbest}^i, in i-th local store membrane by communication rules;
  end for
  Update global best object, O_{gbest}, in global store membrane by communication rules;
  Set computing step $t = 0$;
  Step2: /∗Object evolution in evolution membranes∗/
  for each evolution membrane, i, $(i = 1, \ldots, q)$ in parallel do
      for $j = 1$ to m
         Evolve objects, O_j^i, using the improved differential evolution rules;
         Calculate partition matrix, U_j^i;
         Calculate the XB index of object O_j^i;
      end for
  end for
  Step3: /∗Object communication∗/
  for each evolution membrane, i, $(i = 1, \ldots, q)$, in parallel do
      Transmit the best object in membrane i to update its local best object, O_{lbest}^i, using communication rules;
      Transmit the best object in membrane i to update the global best object, O_{gbest}, using communication rules;
  end for
  step4: /∗Halting condition judgment∗/
  if $t \leqslant t_{max}$ is satisfied
      $t = t + 1$;
      goto Step2;
  end if
  Export the global best object, O_{gbest};
  Calculate partition matrix, U_j^i, according to the global best object, O_{gbest};
  Assign all data points into K clusters based on U_j^i.
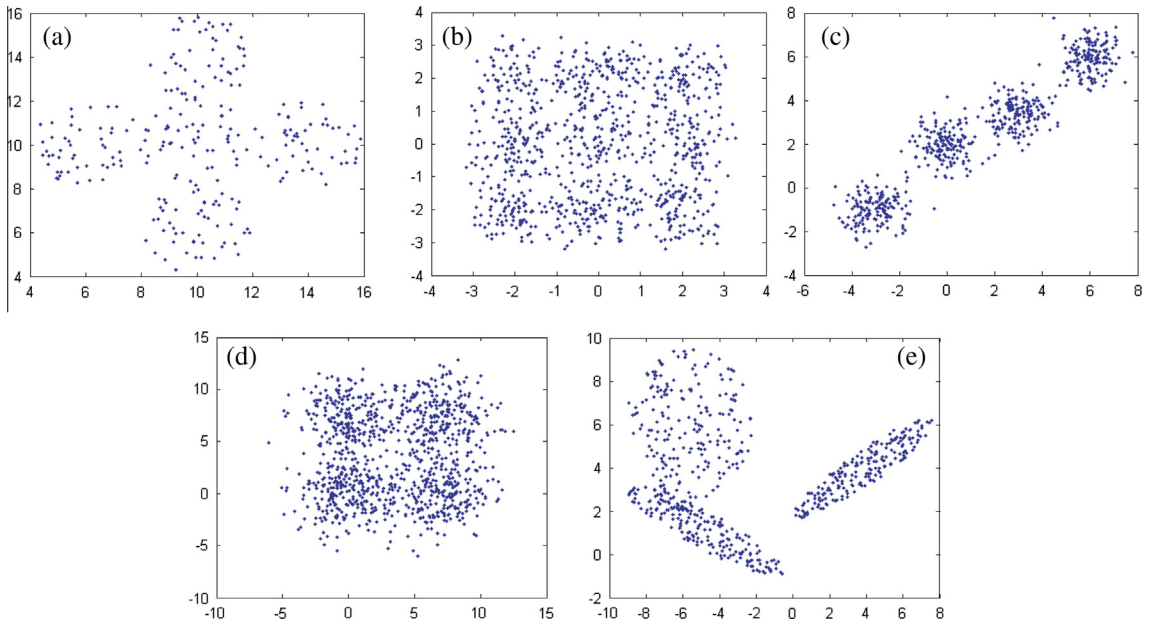End

Fig. 3. Five artificial data sets used in experiments: (a) Data 1; (b) Data 2; (c) Data 3; (d) Data 4; and (e) Data 5.

- Data 4: A two-dimensional data set with four clusters and 1000 points (Fig. 3(d)). Clusters are circular with centers approximately on a square grid.
- Data 5: A two-dimensional data set with three clusters and 600 points (Fig. 3(e)). Two clusters are circular while the third cluster is elliptical.

5.2. Real-life data sets

Five real-life data sets were obtained from the UCI Machine Learning Repository [43].

- Iris: 150 data points distributed over three clusters (Setosa, Versicolor, and Virginica), with four dimensions (sepal length, sepal width, petal length, and petal width).
- Wine: 178 data points with 13 features from a chemical analysis of wines grown in the same region in Italy, from three different cultivators. It is divided into three clusters.
- Glass: 214 data points divided into six clusters. Data dimensions are: id number[A5], Refractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron.
- BreastCancer: 683 points with nine features. There are two categories in the data: malignant and benign, which are known to be linearly separable.
- Newthyroid: 215 data points with five attributes. The original database from where it has been collected is titled as thyroid gland data (normal, hypo and hyper functioning). There are three categories in the data: euthyroidism, hypothyroidism and hyperthyroidism.

5.3. Input parameters

The parameters of the proposed evolution–communication P system were set to: $q = 16$, $m = 20$, $t_{max} = 100$ and $C_r = 0.8$.

The Improved-DE algorithm first algorithm uses the improved differential evolution mechanism, while the Original-DE algorithm uses the original differential evolution mechanism. Parameters for both DE-based algorithms were: population size $PS = 50$, the number of generations $T = 100$, $C_r = 0.8$, and α and F used the same values as for the Fuzzy-CM method.

The GA algorithm was executed for 100 generations with population size 20, crossover probability 0.8, and mutation probability 0.01. The FCM algorithm was executed until it converged to a final solution.

Table 2
The experimental results on five artificial data sets.

Data sets	Methods	XB	PBMF	J_m	FM
Data 1	Fuzzy-MC	0.112 (±0.005)	2.85 (±1.004)	916.49 (±6.623)	0.921 (±0.023)
	Improved-DE	0.235 (±0.009)	2.83 (±1.014)	919.32 (±6.685)	0.827 (±0.031)
	Original-DE	0.793 (±0.023)	2.79 (±1.053)	928.11 (±6.735)	0.784 (±0.033)
	GA	1.653 (±0.187)	2.27 (±1.316)	989.52 (±8.923)	0.753 (±0.041)
	FCM	1.982 (±0.254)	2.23 (±1.523)	996.42 (±10.527)	0.669 (±0.043)
	AL	2.899	1.85	1085.03	0.631
Data 2	Fuzzy-MC	0.143 (±0.018)	0.28 (±0.115)	822.04 (±4.762)	0.725 (±0.019)
	Improved-DE	0.257 (±0.018)	0.28 (±0.118)	822.08 (±4.769)	0.649 (±0.032)
	Original-DE	0.699 (±0.022)	0.26 (±0.122)	822.10 (±4.833)	0.623 (±0.031)
	GA	1.832 (±0.201)	0.24 (±0.238)	823.44 (±5.215)	0.598 (±0.040)
	FCM	2.113 (±0.239)	0.22 (±0.253)	824.21 (±5.739)	0.501 (±0.041)
	AL	2.914	0.17	826.21	0.475
Data 3	Fuzzy-MC	0.100 (±0.000)	11.29 (±4.800)	2023.71 (±28.720)	0.962 (±0.021)
	Improved-DE	0.183 (±0.006)	11.43 (±4.867)	2035.23 (±28.818)	0.946 (±0.033)
	Original-DE	0.762 (±0.023)	11.87 (±4.924)	2041.52 (±29.226)	0.934 (±0.032)
	GA	2.135 (±0.205)	12.21 (±5.844)	2055.18 (±31.004)	0.917 (±0.039)
	FCM	2.512 (±0.268)	12.75 (±6.733)	2061.83 (±30.982)	0.885 (±0.044)
	AL	3.443	15.32	2132.36	0.862
Data 4	Fuzzy-MC	0.100 (±0.000)	80.88 (±8.048)	12842.28 (±25.686)	0.832 (±0.022)
	Improved-DE	0.112 (±0.006)	80.84 (±8.086)	12845.46 (±25.722)	0.751 (±0.029)
	Original-DE	0.157 (±0.034)	79.18 (±8.224)	12876.05 (±26.136)	0.714 (±0.034)
	GA	2.532 (±0.178)	78.67 (±9.336)	12932.11 (±28.322)	0.685 (±0.043)
	FCM	2.483 (±0.245)	78.54 (±9.793)	12942.38 (±25.286)	0.634 (±0.042)
	AL	3.387	75.27	13127.51	0.627
Data 5	Fuzzy-MC	0.1 (±0.0001)	138.93 (±10.558)	11143.86 (±25.210)	0.928 (±0.021)
	Improved-DE	0.103 (±0.005)	138.69 (±10.632)	11145.21 (±25.432)	0.922 (±0.032)
	Original-DE	0.127 (±0.023)	137.54 (±10.847)	11147.05 (±25.917)	0.882 (±0.032)
	GA	0.997 (±0.894)	135.23 (±12.427)	11223.56 (±26.527)	0.857 (±0.042)
	FCM	1.263 (±0.936)	134.72 (±12.369)	11318.46 (±26.982)	0.729 (±0.043)
	AL	2.884	130.80	11452.38	0.715

5.4. Results

Note that except for AL, the clustering algorithms contain some stochastic and/or random factors. Thus, the five clustering algorithms were executed 100 times on each data set and mean and standard variances were calculated for each algorithm over the 100 runs for the XB index, PBMF index, J_m, and F-Measure. Tables 2 and 3 show the results of the clustering algorithms for the artificial and real-life data sets, respectively. Fuzzy-MC and Improved-DE show significant advantages compared with the other algorithms across all four indexes and all data sets. Generally, the clustering performance of Fuzzy-MC is comparable or superior to Improved-DE for both artificial and real data sets, with both of these algorithms being superior to the others. A more comprehensive statistical analysis of the ranking is given below.

5.5. Stability

Fuzzy-MC shows significantly smaller standard variance (see Tables 2 and 3) than the other algorithms for all metrics and all data sets, demonstrating the superiority of the proposed Fuzzy-MS algorithm in terms of stability.

Table 3
The experimental results on five real-life data sets.

Data sets	Methods	XB	PBMF	J_m	FM
Iris	Fuzzy-MC	0.100 (±0.000)	10.79 (±3.722)	413.32 (±10.127)	0.792 (±0.021)
	Improved-DE	0.523 (±0.021)	9.00 (±3.913)	433.95 (±15.332)	0.782 (±0.029)
	Original-DE	0.832 (±0.082)	8.03 (±4.531)	456.12 (±19.223)	0.763 (±0.031)
	GA	0.945 (±0.183)	6.92 (±4.454)	481.32 (±22.657)	0.747 (±0.038)
	FCM	2.025 (±0.314)	5.25 (±4.992)	526.44 (±25.453)	0.713 (±0.042)
	AL	3.253	2.51	568.36	0.694
Wine	Fuzzy-MC	0.1 (±0.000)	25436.18 (±30.889)	3229788.94 (±39.521)	0.625 (±0.022)
	Improved-DE	0.316 (±0.019)	25419.26 (±30.893)	3229803.23 (±39.529)	0.584 (±0.032)
	Original-DE	0.435 (±0.085)	25401.87 (±30.936)	3229812.65 (±39.530)	0.526 (±0.035)
	GA	0.843 (±0.248)	25392.33 (±32.335)	3229869.45 (±40.441)	0.498 (±0.037)
	FCM	1.336 (±0.279)	25383.21 (±33.651)	3229953.36 (±45.223)	0.442 (±0.041)
	AL	4.323	25379.51	3229854.39	0.425
Glass	Fuzzy-MC	0.101 (±0.003)	18.65 (±7.116)	2157.39 (±20.194)	0.481 (±0.021)
	Improved-DE	0.246 (±0.014)	17.97 (±7.120)	2158.91 (±20.832)	0.468 (±0.031)
	Original-DE	0.381 (±0.069)	17.45 (±7.127)	2160.29 (±21.003)	0.451 (±0.033)
	GA	0.752 (±0.313)	16.77 (±8.045)	2162.55 (±22.425)	0.429 (±0.039)
	FCM	2.0738 (±0.298)	16.31 (±9.362)	2162.69 (±23.337)	0.414 (±0.043)
	AL	3.324	15.99	2164.12	0.403
BreastCancer	Fuzzy-MC	0.1 (±0.0001)	201.81 (±7.094)	111409.64 (±34.584)	0.915 (±0.023)
	Improved-DE	0.215 (±0.015)	201.73 (±7.113)	111411.25 (±34.708)	0.861 (±0.028)
	Original-DE	0.471 (±0.052)	200.38 (±7.547)	111436.85 (±35.212)	0.838 (±0.032)
	GA	0.937 (±0.215)	199.43 (±8.454)	111487.38 (±35.834)	0.793 (±0.041)
	FCM	1.997 (±0.553)	199.39 (±9.017)	111533.24 (±37.992)	0.736 (±0.043)
	AL	2.743	197.81	111629.51	0.724
Newthyroid	Fuzzy-MC	0.100 (±0.000)	1590.16 (±13.722)	123181.83 (±21.477)	0.827 (±0.022)
	Improved-DE	0.235 (±0.008)	1589.58 (±13.723)	123183.41 (±21.732)	0.772 (±0.031)
	Original-DE	0.406 (±0.025)	1588.73 (±13.825)	123228.93 (±22.223)	0.746 (±0.036)
	GA	0.862 (±0.194)	1588.28 (±14.653)	123352.47 (±23.527)	0.729 (±0.043)
	FCM	1.556 (±0.472)	1588.39 (±15.268)	123383.32 (±23.465)	0.662 (±0.045)
	AL	2.912	1586.86	123519.07	0.648

Table 4
Significance test (Wilcoxon's rank sum) for Fuzzy-MC against other methods.

Data sets	Improved-DE	Original-DE	GA	FCM	AL
Data 1	0.0138	1.23e−3	3.37e−3	1.79e−4	2.36e−5
Data 2	0.0328	1.87e−3	4.56e−3	1.58e−4	2.28e−5
Data 3	0.0235	1.64e−3	4.47e−3	1.45e−4	2.53e−5
Data 4	0.0272	1.39e−3	4.24e−3	1.43e−4	2.19e−5
Data 5	0.0126	1.16e−3	3.27e−3	1.39e−4	2.35e−5
Iris	0.0482	2.48e−3	5.82e−3	1.83e−4	3.72e−5
Wine	0.0279	1.24e−3	3.99e−3	2.81e−4	2.33e−5
Glass	0.0434	3.19e−3	4.97e−3	1.53e−4	3.61e−5
BreastCancer	0.0318	1.45e−3	4.15e−3	2.36e−4	2.52e−5
Newthyroid	0.0231	1.39e−3	4.24e−3	1.64e−4	2.16e−5

5.6. Statistical significance

The Wilcoxon's rank sum test is a nonparametric statistical significance test for independent samples, and was conducted at the 5% significance level on the experimental outcomes. We created six groups for each data set, corresponding to the six clustering methods. Each group consisted of the F-Measure values from the 100 consecutive runs of the corresponding methods for the data sets. Table 4 shows the p-values generated by Wilcoxon's rank sum test comparing Fuzzy-MC and the other method as indicated. The null hypothesis is that there is no significant difference between mean values of two groups, with the appropriate alternative hypothesis. All p-values are less than 0.05, which establishes that the null hypothesis does not hold for any comparison at the 95% level of confidence. Hence, we have established the statistically significant superiority of the proposed Fuzzy-MC algorithm.

6. Conclusions

This paper proposed a fuzzy membrane clustering algorithm for fuzzy clustering, based on an evolution–communication P system with a modified evolution mechanism corresponding to its special membrane structure and inherent communication mechanism. The proposed P system exploits the optimal cluster centers for a data set and determines good fuzzy partitioning. The proposed fuzzy membrane clustering algorithm was compared with two DE-based fuzzy clustering, GA-based fuzzy clustering, and FCM and hierarchical average linkage clustering. Statistical significance was explicitly tested and established the advantages of the proposed algorithm. Further research is required to apply membrane computing to solve automatic and multiobjective clustering problems.

References

[1] A. Alhazov, On determinism of evolution–communication P systems, J. Univ. Comput. Sci. 10 (5) (2004) 502–508.
[2] S. Bandyopdhyay, S. Saha, GAPS: a clustering method using a new point symmetry-based distance measure, Pattern Recogn. 40 (2007) 3430–3451.
[3] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, New York, 1981.
[4] M. Cavaliere, Evolution–communication P systems, in: WMC-CdeA 2002, LNCS 2597, 2003, pp. 134–145.
[5] D. Chang, X. Zhang, C. Zheng, A genetic algorithm with gene rearrangement for k-means clustering, Pattern Recogn. 42 (2009) 1210–1222.
[6] S. Das, A. Konar, Automatic image pixel clustering with an improved differential evolution, Appl. Soft Comput. 9 (2009) 226–236.
[7] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31.
[8] B. Everitt, S. Landau, M. Leese, Cluster Analysis, Arnold, London, 2001.
[9] R. Freund, Gh. Păun, M.J. Pérez-Jiménez, Tissue-like P systems with channel-states, Theoret. Comput. Sci. 330 (1) (2005) 101–116.
[10] M. Gheorghe, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, Research frontiers of membrane computing: open problems and research topics, Int. J. Found. Comput. Sci. 24 (2013) 547–C623.
[11] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.
[12] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fundam. Inform. 71 (2-3) (2006) 279–308.
[13] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recogn. Lett. 31 (2010) 651–666.
[14] Y.T. Kao, E. Zahara, I.W. Kao, A hybridized approach to data clustering, Expert Syst. Appl. 34 (3) (2008) 1754–1762.
[15] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, Appl. Soft Comput. 11 (2011) 652–657.
[16] R. J Kuo, Y.J. Syu, Z.Y. Chen, F.C. Tien, Integration of particle swarm optimization and genetic algorithm for dynamic clustering, Inf. Sci. 195 (2012) 124–140.
[17] M. Laszlo, S. Mukherjee, A genetic algorithm that exchanges neighboring centers for k-means clustering, Pattern Recogn. Lett. 28 (2007) 2359–2366.
[18] U. Maulik, S. Bandyopadhyay, Genetic algorithm based clustering technique, Pattern Recogn. 33 (2000) 1455–1465.
[19] U. Maulik, S. Bandyopadhyay, Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, IEEE Trans. Geosci. Remote Sens. 41 (5) (2003) 1075–1081.
[20] U. Maulik, I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recogn. 42 (2009) 2135–2149.
[21] P. Melin, J. Amezcua, F. Valdez, O. Castillo, A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias, Inf. Sci. 279 (2014) 483–497.
[22] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, M. Valdez, Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic, Expert Syst. Appl. 40 (8) (2013) 3196–3206.
[23] C.A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, Pattern Recogn. Lett. 17 (1996) 825–832.
[24] C.D. Nguyen, K.J. Cios, GAKREM: a novel hybrid clustering algorithm, Inf. Sci. 78 (2008) 4205–4227.
[25] T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, Appl. Soft Comput. 10 (2010) 183–197.
[26] T.Y. Nishida, An approximate algorithm for NP-complete optimization problems exploiting P systems, in: Proceedings of the Workshop on Uncertainty in Membrane Computing, Palma de Mallorca, 2004, pp. 185–192.
[27] L. Pan, X. Zeng, Small universal spiking neural P systems working in exhaustive mode, IEEE Trans. Nanbiosci. 10 (2) (2011) 99–105.
[28] Gh. Păun, Computing with membranes, J. Comput. Syst. Sci. 61 (1) (2000) 108–143.
[29] Gh. Păun, Tracing some open problems in membrane computing, Romanian J. Inform. Sci. Technol. 10 (4) (2007) 303–314.
[30] Gh. Păun, Twenty six research topics about spiking neural P systems, in: Proc. Fifth Brainstorming Week on Membrane Computing, January 2007, pp. 263–280.
[31] Gh. Păun, M.J. Pérez-Jiménez, Membrane computing: brief introduction, recent results and applications, BioSystems 85 (2006) 11–22.
[32] Gh. Păun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, New York, 2010.

[33] H. Peng, J. Wang, M.J. Pérez-Jiménez, P. Shi, A novel image thresholding method based on membrane computing and fuzzy entropy, J. Intell. Fuzzy Syst. 24 (2) (2013) 229–237.
[34] H. Peng, J. Wang, M.J. Pérez-Jiménez, A. Riscos-Núñez, The framework of P systems applied to solve optimal watermarking problem, Signal Process. 101 (2014) 256–265.
[35] H. Peng, J. Wang, M.J. Pérez-Jiménez, H. Wang, J. Shao, T. Wang, Fuzzy reasoning spiking neural P system for fault diagnosis, Inf. Sci. 235 (2013) 106–116.
[36] S. Saha, S. Bandyopadhyay, A new point based fuzzy clustering technique, Inf. Sci. 179 (2009) 3230–3240.
[37] S. Saha, U. Maulik, Incremental learning based multiobjective fuzzy clustering for categorical data, Inf. Sci. 267 (2014) 35–57.
[38] M.A. Sanchez, O. Castillo, J.R. Castro, An analysis on the intrinsic implementation of the principle of justifiable granularity in clustering algorithms, Recent Adv. Hybrid Intell. Syst. SCI451 (2013) 121–134.
[39] M.A. Sanchez, O. Castillo, J.R. Castro, P. Melin, Fuzzy granular gravitational clustering algorithm for multivariate data, Inf. Sci. 279 (2014) 498–511.
[40] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Anal. Chim. Acta 509 (2) (2004) 187–195.
[41] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic strategy for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[42] M. Tu, J. Wang, H. Peng, P. Shi, Fault diagnosis model of power systems based on adaptive fuzzy spiking neural P systems, Chinese J. Electron. 23 (1) (2014) 87–92.
[43] UCI datasets. <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[44] J. Wang, P. Shi, H. Peng, M.J. Pérez-Jiménez, T. Wang, Weighted fuzzy spiking neural P systems, IEEE Trans. Fuzzy Syst. 21 (2) (2013) 209–220.
[45] J. Wang, L. Zhou, H. Peng, G.X. Zhang, An extended spiking neural P system for fuzzy knowledge representation, Int. J. Innovative Comput. Inform. Control 7 (7A) (2011) 3709–3724.
[46] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 841–847.
[47] R. Xu, D. Wunsch, Survey of clustering algorithm, IEEE Trans. Neural Networks 16 (3) (2005) 645–678.
[48] X. Zeng, L. Xu, X. Liu, L. Pan, On languages generated by spiking neural P systems with weights, Inf. Sci. 278 (2014) 423–433.
[49] G. Zhang, J. Cheng, M. Gheorghe, Q. Meng, A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, Appl. Soft Comput. 13 (3) (2013) 1528–1542.
[50] G. Zhang, M. Gheorghe, L. Pan, M.J. Pérez-Jiménez, Evolutionary membrane computing: a comprehensive survey and new results, Inf. Sci. 279 (2014) 528–551.
[51] X. Zhang, Y. Liu, B. Luo, L. Pan, Computational power of tissue P systems for generating control languages, Inf. Sci. 278 (2014) 285–297.
[52] Z. Zhang, X. Yi, H. Peng, A novel framework of tissue membrane systems for image fusion, Bio-Med. Mater. Eng. 24 (6) (2014) 3259–3266.