An automatic clustering algorithm inspired by membrane computing

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

[a] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[b] School of Electrical and Information Engineering, Xihua University, Chengdu 610039, China
[c] School of Electrical and Electronic Engineering, University of Adelaide, Adelaide 5005, Australia
[d] Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, University of Seville, Sevilla 41012, Spain

Keywords:
Membrane computing
Membrane systems
Tissue-like membrane systems
Automatic clustering Membrane clustering algorithm

A B S T R A C T

Membrane computing is a class of distributed parallel computing models. Inspired from the structure and inherent mechanism of membrane computing, a membrane clustering algorithm is proposed to deal with automatic clustering problem, in which a tissue-like membrane system with fully connected structure is designed as its computing framework. Moreover, based on its special structure and inherent mechanism, an improved velocity-position model is developed as evolution rules. Under the control of evolution-communication mechanism, the tissue-like membrane system cannot only find the most appropriate number of clusters but else determine a good clustering partitioning for a data set. Six benchmark data sets are used to evaluate the proposed membrane clustering algorithm. Experiment results show that the proposed algorithm is superior or competitive to three state-and-the-art automatic clustering algorithms recently reported in the literature.

1. Introduction

Data clustering as one of the most useful data mining techniques has been widely used in many fields, such as pattern recognition, image processing, web mining and biology [7,10,22,38]. Clustering will accomplish such a task that finds out the natural partition from a data set such that data points belonging to the same class are as similar to each other as possible whereas data points from two different classes share the maximum difference [13]. Partional clustering is a class of the most important clustering methods, which attempts to directly decompose the data set into several disjointed clusters according to some criteria [44]. The criteria commonly adopted in clustering is minimizing some measure of dissimilarity in the samples within each cluster and maximizing the disimilarity of different clusters. K-means is a widely used partitional clustering algorithm [16]. However, k-means has the following disadvantages: (1) it is sensitive to the initial cluster centers and easy to get stuck at the local optimal solutions; (2) it takes large time cost to find the global optimal solution when the number of data points is large; (3) it requires a priori specification of the number of clusters.

In recent years, a number of global optimization methods have been introduced to overcome the disadvantages of k-means, such as genetic algorithms (GA), simulated annealing (SA), ant colony optimization (ACO), particle swarm optimization (PSO) and differential evolution (DE) algorithm. The global search ability of the GA was first developed to find the optimal cluster centers for a data set [21]. The GA-based methods use two different coding schemes to express the clustering solutions: (i) using the chromosome directly to encode the cluster number that each data point belongs to [20]; (ii) using the chromosome to describe the cluster centers [2]. First scheme can suffer from huge searching space and high computing cost when the number of data points is very large. Thus, second scheme is commonly adopted by most of GA-based methods [3,19]. Although many GAs have shown good performance for finding the promising regions of the search space, most of them often have two drawbacks: premature convergence and lack of good local search ability. Thus, in order to overcome the problems above, other global searching techniques have been successively developed for data clustering problem. A PSO-based clustering method has been proposed in [17], where the PSO is used to find the optimal cluster centers. In [39], ACO has been introduced to process data clustering problem. Moreover, ACO and SA has been combined to solve clustering problem in [25], and a hybrid evolutionary algorithm based on PSO and ACO to find the optimal cluster centers has been also presented in [24]. In addition, a hybrid clustering method, which combines GA and EM (expectation maximization) to automatically determine the optimal cluster centers has been proposed in [23].

The clustering methods described above use the different global searching techniques to find the optimal cluster centers for a data set to be clustered. However, these clustering methods have a limit in practical application: the number of clusters needs to be

determined a priori. In fact, it is difficult to specify the number of clusters in advance for most application. Thus, it becomes a challenge in such a situation in order to determine an appropriate number of clusters and provide a good partitioning for a data set automatically, that is, automatic clustering problem. In recent years, GA, PSO and DE have been used to deal with the automatic clustering problem. A clustering method that uses the GA to automatically evolve the clusters has been presented in [1]. This method uses a variable-length chromosome to express both the cluster centers and the number of clusters, and then achieves automatic clustering by evolving the two parts at the same time. In [27], a PSO-based automatic clustering method has been reported, which first uses the PSO to find the optimal number of clusters and then determines the corresponding cluster centers by using k-means algorithm. In addition, a variable-length GA to solve automatic fuzzy clustering problem has been developed in [37], while an automatic clustering algorithm based on an improved differential evolution has been presented in [6].

Membrane computing, as a class of distributed parallel computing models, is inspired from the structure and functioning of living cells as well as the cooperation of cells in tissues, organs and populations of cells [35,36]. The models are commonly called membrane systems or P systems. Over the past years, a variety of variants of membrane systems have been proposed [12,15,32–34,40,42,43], including membrane algorithms of solving the global optimization problems. In recent years, membrane algorithms have attracted much attention on applications of membrane computing [26]. The research results on a lot of global optimization problems have shown that compared to the existing evolutionary algorithms, membrane algorithms offer a more competitive method due to three advantages: better convergence, stronger robustness and better balance between exploration and exploitation [14,29–31,45]. Based on the above consideration, this paper proposes an automatic clustering algorithm that uses a tissue-like membrane system with fully connected structure to determine the most appropriate number of clusters and find a good partition for a data set to be clustered. Moreover, a modification of velocity-position model is developed according to its special structure and evolution-communication mechanism, which can accelerate the object evolution and enhance the diversity of objects in the system.

The rest of this paper is organized as follows. Section 2 states the problem to be solved and then presents a brief of introduction of tissue-like membrane systems. The proposed membrane clustering algorithm is described in detail in Section 3. Experimental results and analysis are provided in Section 4. Finally, Section 5 draws the conclusions.

2. Preliminaries

2.1. Data clustering problem

Data clustering in a D-dimensional Euclidean space is such a process, which partitions a data set consisted of n data points into K groups (clusters) according to some similarity measure. It is well-known that minimizing some similarity measure to find the natural partitioning on a non-uniform data set is a NP-hard problem essentially [11,22,24].

Assume that $X = \{X_1, X_2, \ldots, X_n\}$ is a data set of n unlabeled data points, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is its ith data point. For the data set X, a partitional clustering algorithm tries to find a partitioning, $\{C_1, C_2, \ldots, C_K\}$, such that the similarity of the data points in the same cluster is maximum and data points from different clusters differ as far as possible.

K-means algorithm is a widely used clustering technique, which attempts to find the optimal cluster centers for determining a good partitioning of a data set. In order to determine the optimal cluster centers, therefore, a data clustering problem can be viewed as an optimization (minimization) problem. The objective function used in k-means is the following total mean square error:

$$J_m(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{K} \sum_{X_j \in C_i} ||X_j - z_i||^2 \tag{1}$$

where $z_1, z_2, \ldots, z_K$ are the cluster centers of the partitioning, $C_1, C_2, \ldots, C_K$, respectively. Note that in k-means each cluster center is the average of samples in the corresponding cluster. However, in most evolutionary clustering algorithms and the proposed algorithm, the cluster center is a representative point of the corresponding cluster and it is often different from the average of samples. Moreover, the cluster centers, as the solutions of an optimization problem, are determined by the evolutionary clustering algorithms.

In recent years several clustering validity indexes have been proposed to evaluate the goodness of partitioning obtained by a clustering algorithm, such as *DI* index [9], *DB* index [8], *PBM* index [28] and *CS* measure [4]. The existing works have shown that J_m index can well capture only hyperspherical shaped clusters. However, data sets may have different shapes, spatial separations, densities and sizes. Compared with other clustering validity indexes, advantage of *CS* measure lies in the effectiveness of dealing with the clusters with different densities and sizes [4,5,7]. The *CS* measure is defined as follows:

$$CS(K) = \frac{\sum_{i=1}^{K} \left[\frac{1}{N_i} \sum_{X_l \in C_i} \max_{X_j \in C_i} ||X_l - X_j|| \right]}{\sum_{i=1}^{K} \left[\min_{1 \le j \le K, j \ne i} ||m_i - m_j|| \right]} \tag{2}$$

where m_i denotes average (vector) of samples in ith cluster and is calculated as follows:

$$m_i = \frac{1}{N_i} \sum_{X_j \in C_i} X_j \tag{3}$$

Generally, the lower *CS* measure under the constrain that $CS > 0$ means that the obtained partition is better, namely, the considered clustering algorithm gains a good clustering performance, and vice versa.

2.2. Tissue-like membrane systems

Tissue-like membrane systems are a kind of variants of membrane systems, which are inspired from the behavior of multiple single-membrane cells evolved in a common environment. A tissue-like membrane system can be logically viewed as a net, in which each cell is regarded as a processor that deals with the objects and communicates them between the cells along the channels assigned in advance. The object processing is completed by evolution rules while object communication is achieved by communication rules. We briefly review the definition and inherent mechanism of tissue-like membrane systems. More detailed descriptions of tissue-like membrane systems can be found in [12,35].

A tissue-like membrane system of degree q is a construct

$$\Pi = (O, w_1, \ldots, w_q, R_1, \ldots, R_q, R', i_0) \tag{4}$$

where

(1) O is a finite non-empty alphabet (of objects);
(2) $w_i (1 \le i \le q)$ is finite set of strings over O, which represents multiset of objects initially present in cell i;
(3) $R_i (1 \le i \le q)$ is finite set of evolution rules in cell i;
(4) R' is finite set of communication rules of the form $(i, u/v, j)$, which represents communication rule between cell i and cell j, $i \ne j, i, j = 1, 2, \ldots, q, u, v \in O^*$;
(5) i_0 indicates the output region of the system.

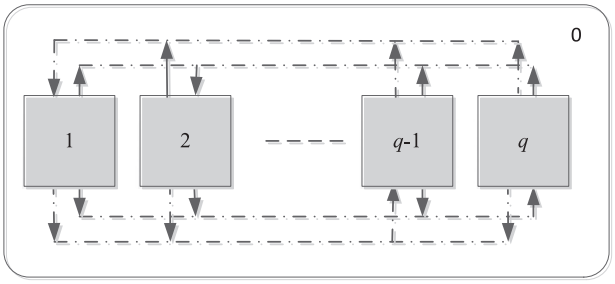

Fig. 1. The designed tissue-like membrane system.

A tissue-like membrane system consists of q cells and each cell is surrounded by a cell membrane, while the region outside the q cells is called the environment. Usually, each cell contains a number of objects. $w_1, w_1, \ldots, w_q$ describe multisets of objects of the q cells, respectively.

In a tissue-like membrane system, there are usually two types of rules: evolution rule and communication rule. Evolution rule is of the form $u \to v$, which means that object u will be evolved to object v. The communication rule between cell i and cell j is of the form $(i, u/v, j)$. The application of this rule means that the objects represented by u and v are interchanged between the two cells. Specially, the communication rule between cell i and the environment is described by the rule $(i, u/\lambda, 0)$, which indicates that object u will transported into the environment.

As usual in the framework of membrane computing, every cell as a computing unit works in a maximally parallel way (a universal clock is considered here). A computation in a tissue-like membrane system is a sequence of computing steps which start with the q cells containing the initial multisets of objects $w_1, \ldots, w_q$ and where, in each step, one or more rules are applied to the current multisets of objects. A computation is successful if and only if it halts. When it halts, it produces a final result in output region.

3. Membrane clustering algorithm for automatic clustering problem

3.1. A tissue-like membrane system designed

The proposed membrane clustering algorithm is an automatic clustering algorithm inspired by inherent mechanism of membrane systems, whose key component is a tissue-like membrane system with fully connected structure, shown in Fig. 1. The tissue-like membrane system consists of q cells that are surrounded by q elementary membranes respectively. Each cell will use the evolution rules to evolve its objects. The dotted lines with direction describe the communication channels between the cells, which are used to achieve the exchange and sharing of objects. The communication of objects are achieved by communication rules. The communication mechanism of objects can realize the co-evolution of objects between the q cells and accelerate the convergence to the global optimum. Assume that the environment (labeled by 0) is the output region of the system.

3.1.1. Objects

The role of the tissue-like membrane system in the proposed membrane clustering algorithm is to determine the most appropriate number of clusters and search for the corresponding optimal cluster centers for a data set. Therefore, each object in cells is used to describe a feasible solution of automatic clustering problem. A maximum number of clusters is assigned a priori, denoted as K_{max}. Thus, the number of clusters of the found good partitioning should be between 2 and K_{max}.

Assume that the data set X to be clustered has at most K_{max} clusters, $C_1, C_2, \ldots, C_{K_{max}}$, and the corresponding cluster centers are

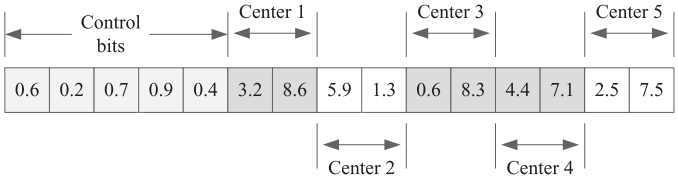

Fig. 2. The example of object representation.

$z_1, z_2, \ldots, z_{K_{max}}$, respectively. Each cluster center is a D-dimensional vector, $z_i = \{z_{i1}, \ldots, z_{iD}\}$, $i = 1, 2, \ldots, K_{max}$. Thus, each object in the system is designed as a $(K_{max} + K_{max} \times D)$-dimensional vector:

$$Z = (\tau_1, \ldots, \tau_{K_{max}}, z_{11}, \ldots, z_{1D}, \ldots, z_{K_{max}1}, \ldots, z_{K_{max}D}) \quad (5)$$

The first K_{max} components, $\tau_1, \tau_2, \ldots, \tau_{K_{max}}$, are real numbers in [0, 1], where component τ_i indicates whether the ith cluster is active, namely, it will participate in classification or not. The remaining parts correspond to K_{max} cluster centers respectively, where $z_{i1}, z_{i2}, \ldots, z_{iD}$ are the D components of ith cluster.

The ith cluster C_i of an object Z is called active cluster if and only if $\tau_i \geq 0.5$. However, if $\tau_i < 0.5$ the cluster is inactive. An active cluster means that it will participate in partitioning data points, otherwise it does not participate in data classification. Therefore, the $\tau_i (i = 1, 2, \ldots, K_{max})$, as some control variables, maintain the selection of active clusters.

Fig. 2 shows an example of object representation. Let the maximum number of clusters, K_{max}, be 5, namely, at most five clusters are assigned a prior, and they are denoted by C_1, C_2, C_3, C_4, C_5, respectively. The centers of the five clusters are (3.2, 8.6), (5.9, 1.3), (0.6, 8.3), (4.4, 7.1) and (2.5, 7.5), respectively. The first five components of the object, (0.6, 0.2, 0.7, 0.9, 0.4), are control variables that indicate whether the corresponding clusters are active. According to the definition of active clusters, C_1, C_3, C_4 are three active clusters. This means that actual clusters expressed by the object are the three clusters, thus the corresponding cluster centers, (3.2, 8.6), (0.6, 8.3) and (4.4, 7.1), will participate in data classification. However, C_2 and C_5 are inactive, so they will be ignored when partitioning data points.

As usual in membrane systems, each cell contains one or more objects and these objects will be evolved by its evolution rules. For simply, assume that the q cells have the same number of objects, denoted by m. During the calculation, the best object found in whole system so far is always stored in the environment, which is called global best object, denoted by Z_{best}.

Initially, the system will generate m initial objects for each cell. When producing an initial object, K_{max} real numbers in [0,1] and $(K_{max} \times D)$ random real numbers that satisfy the ranges of the corresponding component values of samples in data set are generated repeatedly. During object evolution, the system needs to evaluate its each object. The CS measure descried above is used to evaluate the objects or as the fitness values of objects. Generally, the lower the CS measure, the better the object, otherwise, it is worse.

3.1.2. A modification to velocity-position model

The tissue-like membrane system uses evolution rules to evolve the objects in cells. In this work, the velocity-position model in PSO [18] is introduced as the evolution rules of cells. However, a modification of velocity-position model is developed based on the used membrane structure and inherent mechanism of the tissue-like membrane system. The modification is inspired from such an intuition: each cell has two best objects from different sources, first is the best object found by it so far (called local best object, denoted by Z_{lbest}), and another is the best object communicated from other $q-1$ cells (called external best object, which is selected randomly from the $q-1$ communicated best objects, denoted by Z_{ebest}). The two best objects of different sources will participate in or guide the evolution of

objects in cells. The benefit of the consideration is enhancing the diversity of objects in the system and the global exploration ability of the designed tissue-like membrane system.

Assuming that Z_i is ith object in a cell, the modification of velocity-position model can be described as follows:

$$\begin{cases} V_i = wZ_i + c_1r_1(P_i - Z_i) + c_2r_2(Z_{lbest} - Z_i) \\ \qquad\quad + c_3r_3(Z_{ebest} - Z_i) \\ Z_i = Z_i + V_i \end{cases} \tag{6}$$

where P_i is the best position of object Z_i found so far, w is inertia weight, c_1, c_2, c_3 are learning factors, and r_1, r_2, r_3 are three random real numbers in [0,1]. In the implementation, we use the following decreasing strategy of inertia weight:

$$w = w_{max} - (w_{max} - w_{min})t/t_{max} \tag{7}$$

where w_{max} and w_{min} are maximum value and minimum value of inertia weight respectively, and t_{max} is maximum computing step number (or maximum iteration number).

3.1.3. Communication rules

The tissue-like membrane system uses inherent communication mechanism to achieve the exchange and sharing of objects between different cells. The communication mechanism is provided by communication rules. In the tissue-like membrane system, two kinds of communication rules are adopted:

(I) $(i, Z_1Z_2\cdots Z_r/Z'_1Z_1Z'_2\cdots Z'_r, j)$, where $i \neq j$, $i.j = 1, 2, \ldots, q$.
Application of the rule means that r objects $Z_1Z_2\cdots Z_r$ in cell i will be communicated into cell j, and at the same time r objects $Z'_1Z'_2\cdots Z'_r$ in cell j will be communicated into cell i.

(II) $(i, Z_{lbest}/\lambda, 0)$, $i = 1, 2, \ldots, q$.
Here, Z_{lbest} is local best object of cell i in current computing step and λ is an empty object. Application of the rule means that object Z_{lbest} in cell i will be communicated into the environment.

The q cells in the tissue-like membrane system apply first communication rule (I) to establish the communication relationship of objects between them (see the dotted lines in Fig. 1). Each cell transports its best object into other $q-1$ cells by using first communication rule, thus, each cell will receive $q-1$ best objects from other cells. The received $q-1$ best objects constitute a subset of external best objects in the cell, which will participate in object evolution in next computing step. At the same time, each cell communicates its best object into the environment by using second communication rule (II) and updates the global best object Z_{gbest}.

3.1.4. Halting and output

The designed tissue-like membrane system adopts a simple halting condition, namely, maximum computing step number. The tissue-like membrane system will continue to execute until the halting condition is reached, thus, the system halts. When the system halts, the global best object stored in the environment is regarded as final computing result, namely, the determined number of clusters and the corresponding cluster centers.

3.2. Automatic membrane clustering algorithm

The role of the tissue-like membrane system in the developed automatic membrane clustering algorithm is determining the most appropriate number of clusters and find the corresponding optimal cluster centers for a data set. For a data set of n data points, X, the maximum number of clusters, K_{max}, is assigned by a prior. Each object in the tissue-like membrane system represents a vector associated with a group of feasible clusters, including the control variables of active clusters and the corresponding cluster centers.

```
Begin
  Initialize the objects in cells;
  t ← 0;
  While (t < t_max) do
    For each cell in parallel do
      Evolve its objects by using the evolution rules;
      Transport its best object into other cells by the communication rule of type I;
      Update the global best object by the communication rule of type II;
    End
    t ← t + 1;
  End
  Export the number of clusters and the corresponding cluster centers;
  Partition n data points into the corresponding clusters;
End
```

Fig. 3. Pseudo code of the proposed membrane clustering algorithm.

The membrane clustering algorithm first generates m initial objects for each cell, and then executes the tissue-like membrane system. As usual, all cells in the tissue-like membrane system as computing units run in parallel. Each cell uses the improved velocity-position rule to evolve its objects, and then transports its best object into other cells and updates the global best object Z_{gbest}. The evolution-communication procedure is repeated constantly until the halting condition is reached. At this time, the global best object Z_{gbest} is found most appropriate number of clusters, K, as well as the corresponding optimal cluster centers. Finally, the membrane clustering algorithm achieves data clustering by partitioning n data points into K clusters according to the obtained optimal cluster centers. The proposed membrane clustering algorithm is summarized in Fig. 3. In the following, we briery discuss its time and storage complexities. The proposed algorithm consists of three main steps: initialization, object evolution-communication and output. From Fig. 3, it can be observed that initialization step contains double loop (q and m times, respectively), so its time complexity is $O(qm)$. For object evolution-communication step, there are triple loop (q, m, and t_{max} times, respectively), therefore, its time complexity is $O(qmt_{max})$. For output step, its time complexity is $O(nK_{max})$. Therefore, the time complexity of the proposed algorithm is $O(qmt_{max} + nK_{max})$. During the computation, the used membrane system needs to store qm objects (it has q cells and each cell has m objects), so the storage complexity of the proposed algorithm is $O(qm)$.

4. Experimental results and analysis

In order to evaluate the performance of the proposed membrane clustering algorithm, six benchmark data sets from [41] are used in experiments, including *Iris*, *Newthyroid*, *Vowel*, *Glass*, *Wine* and *Cancer*. The six data sets are briefly described as follows:

(1) *Iris*. The data set consists of 150 points distributed over three clusters (*setosa*, *versicolor* and *virginica*). Each data has four features: *sepal length*, *sepal width*, *petal length* and *petal width*.
(2) *Newthyroid*. The data set has 215 points along with five features, which are distributed over three clusters: *euthyroidism*, *hypothyroidism* and *hyperthyroidism*.
(3) *Vowel*. The data set has 871 points along with five features and is divided into six clusters.
(4) *Glass*. The data set has 214 points and is divided into six clusters. Each data point has nine features: *rrefractive index*, *sodium*, *magnesium*, *aluminum*, *silicon*, *potassium*, *calcium*, *barium* and *iron*.
(5) *Wine*. The data set has 178 points along with 13 features. It is divided into three clusters.
(6) *Cancer*. The data set consists of 683 points and each pattern has nine features. There are two categories in the data: *malignant* and *benign*.

Table 1
The numbers of clusters estimated by several clustering algorithms.

Data sets	Actual number	*Membrane systems*	*ACDE*	*GCUK*	*DCPSO*
Iris	3	**3.26**	3.28	2.33	2.21
		±0.028	± 0.039	± 0.098	± 0.045
Newthyroid	3	**3.15**	3.22	2.74	3.31
		±0.015	± 0.036	± 0.086	± 0.038
Vowel	6	**6.25**	5.68	5.03	7.18
		±0.022	± 0.065	± 0.023	± 0.035
Glass	6	**6.03**	6.08	5.89	5.81
		±0.009	± 0.074	± 0.018	± 0.019
Wine	3	**3.12**	3.31	2.74	3.47
		±0.014	± 0.042	± 0.025	± 0.029
Cancer	2	**2.02**	2.05	2.12	2.31
		±0.010	± 0.064	± 0.011	± 0.027

Table 2
The F-measures calculated by several clustering algorithms.

Data sets	*Membrane systems*	*ACDE*	*GCUK*	*DCPSO*
Iris	**0.826**	0.821	0.695	0.728
	±0.009	± 0.022	± 0.032	± 0.043
Newthyroid	**0.842**	0.835	0.762	0.785
	±0.011	± 0.023	± 0.033	± 0.039
Vowel	**0.882**	0.874	0.826	0.829
	±0.008	± 0.025	± 0.036	± 0.044
Glass	**0.492**	0.485	0.429	0.433
	±0.012	± 0.024	± 0.031	± 0.038
Wine	0.683	**0.685**	0.523	0.547
	±0.013	± 0.022	± 0.030	± 0.042
Cancer	**0.967**	0.963	0.826	0.854
	±0.009	± 0.021	± 0.033	± 0.040

In experiments, the proposed membrane clustering algorithm is compared with three recently developed automatic clustering algorithms, which are *GCUK* [1], *ACDE* [5] and *DCPSO* [27] algorithms. In order to evaluate the objects in cells or chromosomes in population, *CS* measure is used as fitness function in these clustering algorithms.

The input parameters of tissue-like membrane system in the proposed membrane clustering algorithm are as follows: the number of cells $q = 4$, the number of objects in each cell $m = 50$, and maximum computing step number $t_{max} = 300$. In the improved velocity-position model, input parameters are $w_{max} = 0.9$, $w_{min} = 0.2$ and $c_1 = c_2 = c_3 = 1.0$. In experiments, *ACDE*, *GCUK* and *DCPSO* use the parameters given in original literature. In the implementation of *ACDE*, we use the population size $NP = 200$, $CR_{max} = 1.8$, $CR_{min} = 0.5$ and iteration number t_{max}=300. For *GCUK*, we set the population size $NP = 200$, crossover probability $p_c = 0.8$, mutation probability $p_m = 0.001$ and iteration number t_{max}=300. The parameters of *DCPSO* are as follows: the population size $NP = 200$, $C_1 = C_2 = 1.494$, $P_{ini} = 0.75$ and iteration number t_{max}=300. In addition, assume $K_{max} = 10$ in the clustering algorithms above.

In order to check the clustering qualities of the clustering algorithms, we use F-measure of the whole partitioning and at the same time examine the optimal numbers of clusters determined by the clustering algorithms. Let m_{ij} be the number of points that belong to both cluster i and cluster j; m_i is the total number of points in cluster i. The F-measure of cluster i with respect to class j is defined by:

$$F(i, j) = \frac{(2 \times precision(i, j) \times recall(i, j))}{(precision(i, j) + recall(i, j))} \tag{8}$$

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

$$F = \sum_j \frac{m_j}{m} \max_i F(i, j) \tag{9}$$

For the F-measure, the optimum score is 1; higher scores are considered better than lower scores.

Since the proposed membrane clustering algorithm, *ACDE*, *GCUK* and *DCPSO* contain some random/stochastic factors, the optimal numbers of clusters determined by them as well as the obtained F-measures of the corresponding clustering partitioning on different runs may be different. Therefore, the clustering algorithms are independently executed 50 times on each data set (with different initial objects or initial population), and then calculate the mean values and standard deviations of the obtained optimal numbers of clusters and F-measures respectively.

Table 1 provides the comparison results of the optimal numbers of clusters estimated by the clustering algorithms. Each data set has an actual number of clusters, while Table 1 gives the mean values and standard deviations of the numbers of clusters estimated by the clustering algorithms the 50 times on each data set. It can be seen from Table 1 that mean values of the proposed membrane clustering algorithm (membrane systems) and *ACDE* algorithm are the most close to the actual number of clusters in each data set. For example, for *Iris*, the mean values of the number of clusters estimated by the proposed membrane clustering algorithm and *ACDE* are 3.26 and 3.28, however, that of *GCUK* and *DCPSO* are 2.33 and 2.21 respectively. Note that the actual number of clusters for *Iris* is 3, so this illustrates the ability of the membrane clustering algorithm for estimating the number of clusters outperforms other three clustering algorithms. The results on *Glass* show that the average number of clusters for the proposed membrane clustering algorithm is 6.03 while the average numbers of clusters for *ACDE*, *GCUK* and *DCPSO* are 6.08, 5.89 and 5.81 respectively. The *Glass* actually has six clusters, so the comparison results indicate that the membrane clustering algorithm is the best of all the clustering algorithms for the prediction ability. At the same time it is clear seen that standard deviations of the numbers of clusters obtained by the proposed membrane clustering algorithm is lower than that of other three clustering algorithms, so the membrane clustering algorithm is robust.

The F-measure is used to evaluate the goodness of clustering partitioning generated by a clustering algorithm. Usually, higher F-measure means that the clustering algorithm has a better clustering performance, otherwise, it has a worse performance. Table 2 shows the comparison results of F-measures obtained by all clustering algorithms on the six benchmark data sets. It can be observed from Table 2 that for all data sets, the mean values of F-measures produced by the proposed membrane clustering algorithm and *ACDE* are higher than that of *GCUK* and *DCPSO*. This illustrates that the proposed membrane clustering algorithm and *ACDE* can find the better clustering partitioning. Moreover, in addition to *Wine*, the clustering performances of the proposed membrane clustering algorithm on other five data sets are better than that of *ACDE*. Table 2 clear shows that the proposed membrane clustering algorithm achieves the lowest standard deviation on all data sets. Therefore, compared with other three algorithms, the proposed membrane clustering algorithm has a stronger robustness.

In experiments, when a clustering algorithm is executed a time on a data set, we compute its classification error according to the obtained partitioning and actual partitioning. Then, we further compute the average classification errors and standard deviations of the 50 runs on each data set for the clustering algorithms, respectively. Table 3 reports comparison results of classification errors produced by the clustering algorithms on six benchmark data sets. The results clear show that compared with other three clustering algorithms, the proposed membrane clustering algorithm has lower average classification error and standard deviation. Therefore, the experimental results further demonstrate the effectiveness of the proposed membrane clustering algorithm in solving automatic clustering problem.

Table 3
The classification errors of several clustering algorithms.

Data sets	*Membrane systems*	*ACDE*	*GCUK*	*DCPSO*
Iris	**2.44**	2.56	5.37	4.92
	±0.01	**±0.01**	± 0.03	± 0.03
Newthyroid	**12.26**	12.33	25.47	22.75
	±1.18	± 1.27	± 1.46	± 1.51
Vowel	**38.51**	40.24	112.15	102.61
	±1.01	± 1.02	± 1.16	± 1.12
Glass	**94.65**	98.85	102.39	107.19
	±0.21	± 0.25	± 0.14	± 0.89
Wine	**37.46**	41.12	107.72	102.32
	±0.07	± 0.08	± 1.28	± 1.25
Cancer	**23.92**	25.18	30.42	29.21
	±0.26	± 0.35	± 1.82	± 1.39

Table 4
Comparison of average computing time (second) over 50 runs.

Data sets	*Membrane systems*	*ACDE*	*GCUK*	*DCPSO*
Iris	**11.38**	11.87	12.62	12.45
Newthyroid	**12.19**	12.35	12.93	12.65
Vowel	**15.58**	15.81	16.38	16.19
Glass	**12.57**	12.85	13.42	13.15
Wine	**13.18**	13.36	13.97	13.62
Cancer	**16.47**	16.62	17.38	17.04

The computing time refers to the spending time of an algorithm when it converges to its best objective function value during its a run. Table 4 provides the comparison results of the four algorithms in terms of computing time (second). It can be seen from Table 4 that the proposed membrane clustering algorithm has the smallest average computing time in the four algorithms. Note that the four algorithms have the same computational load because they contain 200 objects (individuals or particles). The comparison results illustrate that the proposed membrane clustering algorithm has the relatively faster convergence.

The experimental results above demonstrate that the proposed membrane clustering algorithm can find the optimal number of clusters and provide a good clustering partitioning for a data set. We further notice that *Iris*, *Newthyroid* and *Vowel* have the lower dimensions (4-, 5- and 5-dimensions respectively) while dimensions of *Glass*, *Wine* and *Cancer* are higher (9-, 13- and 9-dimensions respectively). It can be seen from the experimental results above that the proposed membrane clustering algorithm not only has a good clustering performance on the low dimensional data sets but also achieves the better clustering effects on the higher dimensional data sets. This observation demonstrates that the proposed membrane clustering algorithm has the better scalability.

5. Conclusions

This paper introduces the inherent mechanism of tissue-like membrane systems to solve automatic clustering problem and proposes an automatic clustering algorithm, called membrane clustering algorithm. A tissue-like membrane system with fully connected structure is designed, and a modification of velocity-position model is developed as evolution rules based on its communication mechanism. The tissue-like membrane system can automatically determine the most appropriate number of clusters as well as the corresponding optimal clustering partitioning for a data set. In order to establish the availability and effectiveness of the proposed membrane clustering algorithm in solving automatic clustering problem, it is compared with three recently developed automatic clustering algorithms on six benchmark data sets. The comparison includes three aspects: the estimated number of clusters, F-measure and classification error. The comparison results indicate that the proposed membrane clustering algorithm can effectively determine the most appropriate number of clusters and provide a good clustering partitioning and it is robust. At the same time, the proposed membrane clustering algorithm not only has a good clustering performance on the low dimensional data sets but also achieves the better clustering effects on the higher dimensional data sets.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant nos. 61170030 and 61472328), and Research Fund of Sichuan Science and Technology Project (No. 2015HH0057), China.

References

[1] S. Bandyopadhyay, U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recogn. 35(6) (2002) 1197–1208.
[2] S. Bandyopdhyay, S. Saha, Gaps: a clustering method using a new point symmetry-based distance measure, Pattern Recogn. 40 (2007) 3430–3451.
[3] D. Chang, X. Zhang, C. Zheng, A genetic algorithm with gene rearrangment for k-means clustering, Pattern Recogn. 42 (2009) 1210–1222.
[4] C. Chou, M. Su, E. Lai, A new cluster validity measure and its application to image compression, Pattern Anal. Appl. 7(2) (2004) 205–220.
[5] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved different evolution algorithm, IEEE Trans. Systems, Man Cybern., Part A 38(1) (2008) 218–237.
[6] S. Das, A. Konar, Automatic image pixel clustering with an improved different evolution, Appl. Soft Comput. 9 (2009) 226–236.
[7] S. Das, S. Sil, Kernel-induced fuzzy clustering of image pixels with an improved different evolution algorithm, Inf. Sci. 180 (2010) 1237–1256.
[8] D. Davies, D. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1(2) (1979) 224–227.
[9] J. Dunn, Well separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1974) 95–104.
[10] B. Everitt, S. Landau, M. Leese, Cluster Analysis, Arnold, London, 2001.
[11] E. Falkenauer, Genetic Algorithms and Grouping Problms, Wiley, New York, 1998.
[12] R. Freund, G. Păun, M. Pérez-Jiménez, Tissue-like p systems with channel-states, Theor. Comput. Sci. 330(1) (2005) 101–116.
[13] J. Hartigan, Clustering Algorithm, Wiley, New York, 1975.
[14] L. Huang, I. Suh, A. Abraham, Dynamic mul-objective optimization based on membrane computing for control of time-varying unstable plants, Inf. Sci. 181(11) (2011) 2370–2391.
[15] M. Ionescu, G. Păun, T. Yokomori, Spiking neural p systems, Fundam. Inf. 71(2–3) (2006) 279–308.
[16] A. Jain, Data clustering: 50 years beyond k-means, Pattern Recogn. Lett. 31 (2010) 651–666.
[17] Y. Kao, E. Zahara, I. Kao, A hybridized approach to data clustering, Expert Syst. Appl. 34(3) (2008) 1754–1762.
[18] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Network, 1999, pp. 1942–1948.Piscataway, NJ
[19] M. Laszlo, S. Mukherjee, A genetic algorithm that exchanges neighboring centers for k-means clustering, Pattern Recognit. Lett. 28 (2007) 2359–2366.
[20] U. Maulik, S. Bandyopadhyay, Genetic algorithm based clustering technique, Pattern Recognit. 33 (2000) 1455–1465.
[21] C. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, Pattern Recognit. Lett. 17 (1996) 825–832.
[22] M. Naldi, R. Campello, E. Hruschka, A. Carvalho, Efficiency of evolutionary k-means, Appl. Soft Comput. 11 (2011) 1938–1952.
[23] C. Nguyen, K. Cios, Gakrem: a novel hybrid clustering algorithm, Inf. Sci. 78 (2008) 4205–4227.
[24] 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.
[25] T. Niknam, J. Olamaie, B. Amiri, A hybrid evolutionary algorithm based on aco and sa for cluster analysis, J. Appl. Sci. 8(15) (2008) 2695–2702.
[26] T. 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] M. Omran, A. Salman, A. Engelbrecht, Dynamic clustering using particle swarm optimization with application in unsupervised image classification, Pattern Anal. Appl. 8(4) (2006) 332–344.
[28] M. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognit. Lett. 37(3) (2004) 487–501.
[29] H. Peng, J. Wang, M. Pérez-Jiménez, Optimal multi-level thresholding with membrane computing, Digit. Signal Process. 37 (2015a) 53–64.
[30] H. Peng, J. Wang, M. Pérez-Jiménez, A. Riscos-Núñez, The framework of p systems applied to solve optimal watermarking problem, Signal Process. 101 (2014) 256–265.
[31] H. Peng, J. Wang, M. Pérez-Jiménez, A. Riscos-Núñez, An unsupervised learning algorithm for membrane computing, Inf. Sci. 304 (2015b) 80–91.

[32] H. Peng, J. Wang, M. 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.
[33] H. Peng, J. Wang, M. Pérez-Jiménez, H. Wang, J. Shao, T. Wang, Fuzzy reasoning spiking neural p system for fault diagnosis, Inf. Sci. 235 (2013) 106–116.
[34] G. Păun, M. Pérez-Jiménez, Membrane computing: brief introduction, recent results and applications, BioSystems 85 (2006) 11–22.
[35] G. Păun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, New York, 2010.
[36] G. Păun, Computing with membranes, J. Comput. Syst. Sci. 61(1) (2000) 108–143.
[37] S. Saha, S. andyopadhyay, A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters, Inf. Sci. 179 (2009) 3230–3246.
[38] S. Saha, S. Bandyopadhyay, A symmetry based multiobjective clustering technique for automatic evolution of clusters, Pattern Recognit. 43 (2010) 738–751.
[39] P. Shelokar, V. Jayaraman, B. Kulkarni, An ant colony approach for clustering, Anal. Chim. Acta 509(2) (2004) 187–195.
[41] UCI, Uci datasets [online]. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.
[42] J. Wang, P. Shi, H. Peng, M. Pérez-Jiménez, T. Wang, Weighted fuzzy spiking neural p systems, IEEE Trans. Fuzzy Syst. 21(2) (2013) 209–220.
[43] J. Wang, L. Zhou, H. Peng, G. Zhang, An extended spiking neural p system for fuzzy knowledge representation, Int. J. Innov. Comput. 7(7A) (2011) 3709–3724.
[44] R. Xu, D. Wunsch, Survey of clustering algorithm, IEEE Trans. Neural Netw. 16(3) (2005) 645–678.
[45] G. Zhang, J. Cheng, M. Gheorghe, Q. Meng, A hybrid approach based on different evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, Appl. Soft Comput. 13(3) (2013) 1528–1542.