International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **2**, April **2014**

 ISSN 1349-4198
pp. 753–765

A NOVEL CLUSTERING ALGORITHM BASED ON P SYSTEMS

Yang Jiang[1], Hong Peng[1,*], Xiaoli Huang[1]
Jiarong Zhang[1] and Peng Shi[2,3]

[1]Center of Radio Administrator and Technology Development
Xihua University
Chengdu 610039, P. R. China
ph.xhu@hotmail.com

[2]College of Automation
Harbin Engineering University
Heilongjiang Province, Harbin 150001, China

[3]School of Engineering and Science
Victoria University
Melbourne, Vic 8001, Australia

Received March 2013; revised July 2013

Abstract. *Membrane computing (known as P systems) is a novel class of distributed parallel computing models. In this paper, a partition-based clustering algorithm under the framework of membrane computing is proposed. The clustering algorithm is based on a tissue-like P system, which is used to exploit the optimal cluster centers for a data set. Each object in the tissue-like P system represents a group of candidate cluster centers and is evolved through simulated annealing mechanism and mutation mechanism. Meanwhile, communication rules are used to exchange and share the objects between different elementary membranes and between elementary membranes and the environment. The proposed clustering algorithm is evaluated over two artificial data sets and two real-life data sets and is further compared with k-means algorithm and GA-based k-means algorithm respectively. The comparison results reveal the superiority of the proposed clustering algorithm in terms of clustering quality and stability.*

Keywords: Membrane computing, Tissue-like P systems, Clustering algorithm, Simulated annealing, K-means

1. **Introduction.** Membrane computing initiated by Gh. Păun [1] in 2000 is inspired from the structure and functioning of living cells and the interactions of living cells in tissues or higher order biological structures. Membrane computing is a novel class of distributed parallel computing models, and also is known as P systems. Generally, a P system contains three ingredients: (i) membrane structure, (ii) multisets, and (iii) evolution rules [2]. Multisets of objects are placed in the compartments surrounded by the membranes and evolved by some given evolution rules. Therefore, P systems can process and generate information to accomplish a computation. Cell-like P systems were introduced and studied firstly, where the membranes were arranged as a rooted tree [1]. Neural-like P systems are another type of P systems, in which spiking neural P systems, as a class of neural-like P systems, have been widely studied in recent years [3, 4, 5, 6]. Tissue-like P systems were inspired from intercellular communication and cooperation between cells in tissues [7]. In a tissue-like P system, the communication of objects is based on symport/antiport rules, which are introduced as communication rules of the P systems. In the case of symport rules, objects cooperate to traverse a membrane together in the same direction, whereas in the case of antiport rules, objects residing at both

sides of the membrane cross it simultaneously but in opposite directions. A tissue-like P system can be viewed as a net of processors dealing with symbols and communicating these symbols along channels specified in advance. In recent years, a large number of P systems and variants have been proposed [8, 9, 10, 11, 12]. These efforts have addressed that P systems possess maximum parallelism, synchronization and non-deterministic features.

Clustering is a procedure of grouping objects according to the similarity of objects. Any clustering should have two main characteristics: low inter-class similarity and high intra-class similarity. Clustering is an unsupervised learning, namely, it learns by observation rather than from examples. The overall distribution pattern and correlation among data objects can be discovered by clustering [13]. Clustering analysis has been widely used in various fields, such as image processing, data analysis, market analysis. k-means algorithm is a classical partition-based clustering algorithm, in which starting with k random cluster centers, the centers are updated by the arithmetic means of the points belonging to the corresponding clusters iteratively [14]. k-means algorithm has been widely used because of its simplicity and easy implementation. However, Kanugo et al. [15] stated that the selection of initial cluster centers for k-means has a great impact on clustering results. The clustering quality will be worsened if the selection of initial cluster centers is flawed. In addition, k-means easily falls into a local optimum during clustering procedure. In order to overcome the problems of k-means algorithm, several clustering algorithms based on genetic algorithms (GAs) have been developed in recent years [14, 16, 17]. Maulik and and Bandyopadhyay [14] proposed a genetic algorithm based method to process the clustering problem and experiment on synthetic and real life data sets to evaluate the performance. Bandyopadhyay and Saha [16] described an evolutionary clustering technique based on genetic algorithms that used a new point symmetry-based distance measure. Laszlo and Mukherjee [17] presented a genetic algorithm for selecting centers to seed the popular k-means method for clustering, where a crossover operator was used to exchange neighboring centers.

This paper proposes a novel partition-based clustering algorithm, which is based on a tissue-like P system. Cluster centers are represented by the objects in the elementary membranes. Simulated annealing mechanism and mutation mechanism are introduced as evolution rules to evolve the objects, while communication rules between elementary membranes and between elementary membranes and the environment are used to exchange and share the objects. An adaptive modification strategy for initial temperature is adopted in this system in order to better fit various data sets. The designed tissue-like P system can automatically search for the optimal cluster centers to achieve data clustering. We will compare the presented clustering algorithm with classical k-means algorithm and the clustering technique based on genetic algorithms in terms of both clustering quality and stability.

The main contribution of this paper is presentation of a new partition-based clustering algorithm under the framework of membrane computing to solve the clustering problem, in which the mechanisms of tissue-like P systems are applied to exploit the optimal cluster centers for data clustering.

The rest of this paper is organized as follows. In Section 2, tissue-like P systems with symport/antiport rules are reviewed briefly. The presented clustering algorithm based on tissue-like P systems is described in Section 3 and the experimental results are showed in Section 4. Finally, conclusions are drawn in Section 5.

2. Tissue-Like P Systems with Symport/Antiport Rules.

The proposed clustering algorithm is based on tissue-like P systems with symport/antiport rules, so we briefly

review the tissue-like P systems in this section. The more detailed description of the tissue-like P systems can be found in references [2, 7].

Formally, a tissue-like P system of degree $q > 0$ with symport/antiport rules is a structure of the form

$$\Pi = (w_1, \ldots, w_q, R_1, \ldots, R_q, R', i_0)$$

where

(1) w_i $(1 \leq i \leq q)$ is a finite set of strings, representing the multisets of objects associated with cell i in the initial configuration;
(2) R_i $(1 \leq i \leq q)$ is a finite set of evolution rules contained in cell i;
(3) R' is a finite set of communication rules of the form $(i, u/v, j)$, which represents the communication rule between cell i and cell j, $i \neq j$ and $i, j = 0, 1, 2, \ldots, q$;
(4) i_0 indicates the output region of the system.

A tissue-like P system of degree q can be viewed as a net composed of q cells. The q cells are labeled by $1, 2, \ldots, q$ respectively, while the environment is labeled by 0. The communication rule of the form $(i, u/v, j)$ reflects the synapse connection between cell i and cell j implicitly. Each cell is surrounded by an elementary membrane.

$w_1, w_2, \ldots, w_q$ describe the multisets of objects placed in the q cells, respectively. We assume that any multiset of objects is available in the environment.

R_i is a finite set of evolution rules with the form of $u \to v$, $1 \leq i \leq q$. Object u will be evolved into v if the rule is applied. The communication rules of the form $(i, u/v, j)$ are called antiport rules. The communication rule $(i, u/v, j)$ can be applied over two cells labeled by i and j if u is contained in cell i and v is contained in cell j. The application of this rule means that the multisets of objects represented by u and v are interchanged between the two cells. Note that if either $i = 0$ or $j = 0$ then the objects are interchanged between a cell and the environment. The rules described above with one of i, j being empty are called symport rules, for example, $(i, u/\lambda, j)$. Application of the rule means that object u will be communicated from cell i to cell j.

In tissue-like P systems, each cell is a computing unit (as usual in the framework of membrane computing), working in a maximally parallel way (a universal clock is considered). A computation of a tissue-like P system is a sequence of computing steps, which starts from the q cells containing $w_1, \ldots, w_q$. At each step, one or more rules are applied to the current multisets of objects. When the system halts, the computation is accomplished successfully, and the results are generated in the output cell.

3. The Proposed Clustering Algorithm Based on Tissue-Like P Systems.

3.1. The designed tissue-like P system.

In this work, a tissue-like P system is designed to realize a partition-based clustering algorithm. The tissue-like P system consists of q elementary membranes, which are labeled by $1, 2, \ldots, q$ respectively. Figure 1 shows the membrane structure of the system, where the environment is labeled by 0. As usual,

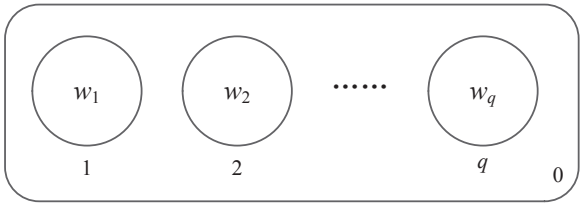

FIGURE 1. The membrane structure of the designed tissue-like P system

each elementary membrane contains one or more objects. In the designed system, each object is a d-dimensional vector, $a = (x_1, x_2, \ldots, x_d) \in R^d$.

Formally, the tissue-like P system of degree q can be described as follows:

$$\Pi = (w_1, \ldots, w_q, R_1, \ldots, R_q, R', i_0)$$

where:

(1) $w_1, \ldots, w_q$ are the initial objects placed in the elementary membranes labeled by $1, \ldots, q$, respectively. Each object is a d-dimensional vector;
(2) R_i is a finite set of evolution rules in cell i, $1 \leq i \leq q$. There are evolution rules of the following two forms:
   (a) Simulated annealing rule: $u \to v$. Simulated annealing mechanism is utilized to evolve object u, generating the new object v;
   (b) Mutation rule: $v \to v'$. A slight mutation on object v is happened to generate object v';
(3) R' is a finite set of communication rules of the form $(i, u/v, j)$, where $i, j \in \{0, 1, \ldots, q\}$ and $i \neq j$, and 0 represents the environment;
(4) $i_0 = 0$ indicates that the environment is the output region.

The communication rule $(i, u/v, j)$ expresses the interchange of object u in cell i and object v in cell j. If $v = \lambda$, object u will be communicated from cell i to cell j and vice versa, where the symbol λ represents the empty object. Assume here that when object u placed in cell i is transported to cell j, a copy of object u will remain in cell i still.

The sum of the Euclidean distances of the data points to their corresponding cluster centers is used as clustering metric to evaluate the objects in the system. Suppose that the data set to be clustered has k cluster centers, $z_1, z_2, \ldots, z_k$, and the corresponding clustering partitions are $C_1, C_2, \ldots, C_k$ respectively. Thus, the clustering metric M is given by

$$M(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \| x_j - z_i \|$$

where $x_1, x_2, \ldots, x_n$ are the data points to be clustered. Generally, the M is less, the clustering result or object is better.

In the designed tissue-like P system, each elementary membrane contains only one object, which represents a group of candidate cluster centers. The cluster centers in the elementary membranes will be evolved into new cluster centers by using simulated annealing rules firstly. The optimal object (cluster centers) found in the computing procedure is always retained in the environment. At the end of each computing step, the best object in each elementary membrane is transported into the environment to update the optimal object by using communication rule. After updating, the updated optimal object is sent back to each elementary membrane to replace the previous object. Then mutation rules are applied in the $q - 1$ elementary membranes except for membrane 1. The optimal object received from the environment takes place slight mutation in the $q - 1$ elementary membranes, and the objects are replaced by the generated new objects. The process described above repeats until the halt condition is satisfied. The object in the environment is the final cluster centers when the system halts.

3.2. **Partitioning and computing new cluster centers.** For k cluster centers $z_1, z_2, \ldots, z_k$, if the distance of a data point x_i to cluster centers z_j $(j = 1, 2, \ldots, k)$ satisfies

$$\| x_i - z_j \| < \| x_i - z_p \|, \text{ for } p = 1, 2, \ldots, k, \text{ and } j \neq p,$$

then the point x_i is assigned to the cluster C_j, $i = 1, 2, \ldots, n$.

After clustering partitions are determined, new cluster centers are computed by the arithmetic means of the points in the corresponding cluster. Therefore, if the number of data points in the k clusters $C_1, C_2, \ldots, C_k$ are $n_1, n_2, \ldots, n_k$ respectively, then the new cluster centers can be computed by

$$z_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$$

where $i = 1, 2, \ldots, k$.

3.3. **Simulated annealing rule.** In the designed tissue-like P system, the objects (cluster centers) in each elementary membrane are evolved by evolution rules, including simulated annealing rules and mutation rules. Simulated annealing rules are inspired from the mechanism of the known simulated annealing algorithm [18, 19].

The simulated annealing rules used in this work are described as follows:

(1) Generate a new solution by adding a perturbation in current partitions, namely, changing the partitions of a or several points randomly;
(2) Compute the difference ΔM between the M value of the solution before the perturbation and the M value of the new solution after the perturbation;
(3) If $\Delta M < 0$, then previous solution is replaced by the new solution;
(4) Otherwise, the new solution is accepted with the probability $e^{-\Delta M/t}$, where t is the temperature;
(5) Repeat steps (1)-(4) until the maximum number of iterations is reached or the solution does not change in a specified number of consecutive iterative steps.

3.4. **Mutation rule.** In each computing step, environment can transport the current optimal object (cluster centers) to the q elementary membranes by using communication rules. And then, the optimal object is mutated in the $q-1$ elemental membranes except membrane 1. The mutated objects will be used as new objects in the next computing step. The mutation rules used in the tissue-like P system can be described as follows.

If the value of a center at dimension j is v, after mutation it becomes

$$v' = \begin{cases} v \pm 0.1 \times \delta \times v, & v \neq 0 \\ v \pm 0.1 \times \delta, & v = 0 \end{cases}$$

where the signs "+" or "−" occur with equal probability, and δ is a real number in the range $[0, 1]$, generated with uniform distribution.

3.5. **Adaptive modification of the initial temperature.** Existing works have indicated that simulated annealing algorithms are sensitive to initial temperature t. Thus, in order to make the proposed clustering algorithm suit various data sets better, an adaptive modification method of initial temperature is introduced into the simulated annealing rules used in this paper.

When $\Delta M \geq 0$, the new solution is accepted with the probability $e^{-\Delta M/t}$ during evolution. Therefore, the acceptance probability of the new solution is associated with the ratio of ΔM and temperature t closely. However, the values of ΔM often have huge differences for different data sets. If the values of the points in the data sets are large, then ΔM is often large, whereas if the values are small, ΔM becomes small.

In order to have a same clustering capability for different data sets, the ratio of ΔM and t should be the same to some extent. Therefore, different initial temperatures need to be explored and set up for different data sets.

If the initial temperatures $t = \beta \Delta M$, the probability is related with the factor β only, having nothing to do with ΔM, namely having nothing to do with data sets. In this way,

it does not need to set up different initial temperatures for different data sets and has the same clustering capability at the same time.

Initially, the temperature t is set to be 1. In the first evolution, the absolute values of ΔM produced in membrane 1 are recorded and their mean value $\Delta M'$ is calculated. Thus, the initial temperature t will be modified as $t = \beta \Delta M'$.

3.6. **Clustering algorithm.** Based on the tissue-like P system, the proposed clustering algorithm can be described as follows:

(1) Generate a group of initial cluster centers (object) for each elementary membrane randomly. And then, determine the clustering partitions according to the cluster centers and compute the new cluster centers. Let initial temperature be 1;
(2) Simulated annealing rules are applied to evolve the cluster centers (objects) in the elementary membranes. If it is the first time of evolution, membrane 1 can adaptively amend the initial temperature according to the evolution results;
(3) The q elementary membranes use communication rules to update the optimal cluster centers in the environment, and then their previous cluster centers are replaced by the updated optimal cluster centers;
(4) The elementary membranes labeled by $2, 3, \ldots, q$ use the mutation rules to generate new objects (cluster centers);
(5) Decrease the temperature by $t = \alpha t$, where $0 < \alpha < 1$;
(6) Repeat steps (2)-(5) until reaching the maximum number of iterations. The system halts and exports the optimal clustering centers in the environment.

4. Experimental Results and Analysis.

4.1. **Experimental data sets.** The clustering algorithm presented in this paper is evaluated over four data sets respectively, including two real-life data sets (*Iris*, *Vowel*) and two artificial data sets (*Data3*, *Data4*). These four data sets are described as follows.

4.1.1. *Real-life data sets.*

- *Iris Data.* The data set represents different categories of irises, which have four features. The four features represent the sepal length, sepal width, petal length and the petal width in centimeters respectively [20]. The data set has three classes with 50 samples per class. There are some overlaps between classes 2 and 3. The number of the cluster centers k is chosen to be 3 for the data set.
- *Vowel Data.* The data set consists of 871 Indian Telugu vowel sounds [21]. They were uttered in a consonant-vowel-consonant context by three male speakers in the age of 30-35 years. The data set has three features, F_1, F_2, F_3, corresponding to the first, second and third vowel formant frequencies, and six overlapping classes $\{\delta, a, i, u, e, o\}$. Therefore, the value of k, representing the number of cluster centers, is chosen to be 6 for this data set.

4.1.2. *Artificial data sets.*

- *Data3*: The data set has nine overlapping classes and all the classes are assumed to have an equal priori probability ($= 1/9$). It has 900 data points with two dimensions, which obey the triangular distribution. The $X - Y$ ranges for the nine classes are as follows:
  Class 1: [–3.3, –0.7] × [0.7, 3.3]
  Class 2: [–1.3, 1.3] × [0.7, 3.3]
  Class 3: [0.7, 3.3] × [0.7, 3.3]
  Class 4: [–3.3, –0.7]× [–1.3, 1.3]

Class 5: [–1.3, 1.3] × [–1.3, 1.3]
Class 6: [0.7, 3.3]× [–1.3, 1.3]
Class 7: [–3.3, –0.7] × [–3.3, –0.7]
Class 8: [–1.3, 1.3]× [–3.3, –0.7]
Class 9: [0.7, 3.3] × [–3.3, –0.7]

Thus, the domain for the triangular distribution for each class and for each axis is 2.6. Consequently, the height will be $\frac{1}{1.3}$ (since $\frac{1}{2} * 2.6 * height = 1$). This data set is shown in Figure 2. The value of k is chosen to be 9 for this data set.

- *Data4*: This is an overlapping data set with ten dimensions generated by a triangular distribution of the form shown in Figure 3 for two classes. It has 1000 data points. The value of k is chosen to be 2 for the data set. The range for class 1 is [0, 2] × [0, 2] × [0, 2] ... 10 times, and that for class 2 is [1, 3]× [0, 2] × [0, 2] ... 9 times, with the corresponding peaks at (1, 1) and (2, 1). The distribution along the first axis x for class 1 may be formally computed by

$$f_1(x) = \begin{cases} 0, & x \le 0, \\ x, & 0 < x \le 1, \\ 2 - x, & 1 < x \le 2, \\ 0, & x > 2. \end{cases}$$

Similarly for class 2

$$f_2(x) = \begin{cases} 0, & x \le 1, \\ x - 1, & 1 < x \le 2, \\ 3 - x, & 2 < x \le 3, \\ 0, & x > 3. \end{cases}$$

The distributions along the other nine axes ($y_i, i = 1, 2, \ldots, 9$) for the both classes are

$$f(y_i) = \begin{cases} 0, & y_i \le 0, \\ y_i, & 0 < y_i \le 1, \\ 2 - y_i, & 1 < y_i \le 2, \\ 0, & y_i > 2. \end{cases}$$

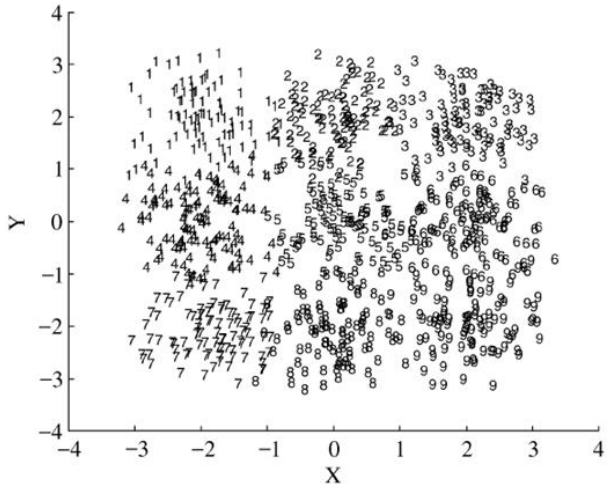

FIGURE 2. Data 3 ('1' – points from class 1, '2' – points from class 2, ..., '9' – points from class 9)

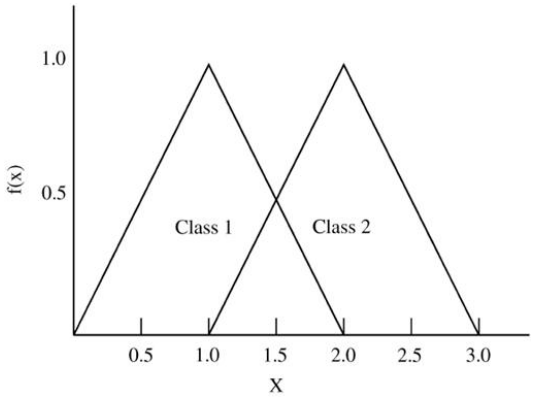

FIGURE 3. Triangular distribution along the X-axis

4.2. **Experimental parameters.** In the experiments, the number of the elementary membranes, q, is chosen to be 4. The maximum cooling times is 1000, and the cooling rate α is 0.99. If the solution does not change in 50 consecutive iterations, the system is considered to be steady and the temperature is reduced. The maximum iteration number is 100 at a temperature. The initial temperature is 1 at first, and then it is modified adaptively by $t = \beta\Delta M'$ for $\beta = 0.24$, at the first iteration. Assume that the clustering partition of one point will be disturbed randomly.

4.3. **Performance comparison and analysis.** The performance of the presented clustering algorithm based on tissue-like P system is compared with two representative clustering algorithms, which are the classical k-means algorithm and genetic algorithm-based clustering algorithm [14] (GA-k-means for short), respectively. Because the clustering algorithm proposed in this paper is a clustering algorithm in the framework of membrane computing, it is called membrane clustering algorithm (MC for short). For the GA-k-means, the parameters of the GA are chosen as follows: the population size is 100, the crossover probability is 0.8, the mutation probability is 0.001 and the number of iterations is 1000.

In the experiments, we independently execute the three clustering algorithms 10 times over each data set, and then obtain their optimal M values. In addition, we compute the mean values and variances 10 times for the three clustering algorithms over each data set. The variances are used to compare the stabilities of the three clustering algorithms, while other results are used to evaluate the clustering qualities of the three clustering algorithms over data sets.

Table 1 shows the experimental results of the three clustering algorithms over *Iris* for 10 times. The comparison results indicate clustering quality of MC is more excellent than that of other two algorithms for every time. Meanwhile, the worst value, best value, average value and variance of MC for 10 runs are all superior to that of k-means and GA-k-means.

The comparison results of the three algorithms over *Vowel* are provided in Table 2. It can be found that the MC in terms of M values is superior to k-means and GA-k-means for every time of 10 runs. In addition, the worst value, best value and mean value of MC are all better than that of other two algorithms obviously. The variance of MC is little worse than that of GA-k-means, but much better than that of k-means.

The comparison results for *Data3* are listed in Table 3. The comparison results show that MC has one bad value and nine stable values that outperform the values of k-means and GA-k-means obviously. In terms of the worst value, best value, mean value

TABLE 1. The comparison results of M values for k-means, GA-k-means and MC over *Iris*

	k-means	GA-k-means	MC
run = 1	97.325924	96.875570	96.662134
run = 2	97.346220	96.929054	96.666562
run = 3	97.346220	96.925511	96.658938
run = 4	97.346220	96.936318	96.657933
run = 5	97.325924	96.920160	96.657690
run = 6	97.346220	97.159438	96.656597
run = 7	97.346220	96.907061	96.661683
run = 8	97.325924	96.927979	96.662329
run = 9	97.346220	96.860877	96.656245
run = 10	97.346220	96.896913	96.664151
Worst	97.346220	97.159438	96.666562
Best	97.325924	96.860877	96.656245
Mean	97.340131	96.933888	96.660426
Variance	0.000096	0.006890	0.000012

TABLE 2. The comparison results of M values for k-means, GA-k-means and MC over *Vowel*

	k-means	GA-k-means	MC
run = 1	151318.839537	149343.289938	149085.792884
run = 2	150397.667727	149297.756410	149086.381404
run = 3	154443.846766	149315.881433	149065.448492
run = 4	149446.887122	149319.662329	148988.994027
run = 5	161006.605287	149326.084239	149085.349820
run = 6	150360.497492	149276.366603	149066.885054
run = 7	158967.269527	149305.861467	149096.103505
run = 8	150469.895346	149248.785820	149088.133764
run = 9	151469.254956	149360.703439	149084.200723
run = 10	149708.671061	149295.692895	148998.086522
Worst	161006.605287	149360.703439	149096.103505
Best	149446.887122	149248.785820	148988.994027
Mean	152758.943482	149309.008457	149064.537620
Variance	16665013.263863	1033.998796	1492.314283

and variance, the performance of MC has a great improvement compared with k-means. Although the worst value of MC is not superior to that of GA-k-means, MC outperforms GA-k-means distinctly at most of times.

Table 4 shows the comparison results over *Data4*. The results indicate that whether the every M value of MC for 10 runs or the worst value, best value and mean value of MC are all better than that of k-means and GA-k-means. Meanwhile, the variance of MC is also less than that of other two algorithms.

Figures 4-7 show the average trends of 10 runs for the three algorithms over the four data sets, respectively. Figure 4 shows the average trend over *Iris*, and Figure 5 shows the average trend over *Vowel*. The average trends for *Data3* and *Data4* are showed in Figure 6 and Figure 7, respectively.

TABLE 3. The comparison results of M values for k-means, GA-k-means and MC over *Data3*

	k-means	GA-k-means	MC
run = 1	603.094514	600.017769	599.760894
run = 2	660.769499	600.016007	599.778260
run = 3	600.538031	600.028848	599.780606
run = 4	603.094514	600.013967	640.861715
run = 5	663.778316	600.077222	599.780754
run = 6	600.756731	600.013897	599.775377
run = 7	644.667212	600.026851	599.782653
run = 8	603.243637	600.022140	599.661831
run = 9	600.402814	600.093488	599.785860
run = 10	600.494550	600.067858	599.781371
Worst	663.778316	600.093488	640.861715
Best	600.402814	600.013897	599.661831
Mean	618.083982	600.037805	603.874932
Variance	724.044847	0.000891	168.893002

TABLE 4. The comparison results of M values for k-means, GA-k-means and MC over *Data4*

	k-means	GA-k-means	MC
run = 1	1256.160089	1256.142306	1256.113775
run = 2	1256.164154	1256.151663	1256.115184
run = 3	1256.164154	1256.151663	1256.112702
run = 4	1256.164154	1256.145110	1256.114176
run = 5	1256.164154	1256.142268	1256.117655
run = 6	1256.160089	1256.153977	1256.117133
run = 7	1256.164154	1256.150406	1256.115940
run = 8	1256.160089	1256.141780	1256.115069
run = 9	1256.164154	1256.140485	1256.114227
run = 10	1256.164154	1256.147206	1256.115322
Worst	1256.164154	1256.153977	1256.117655
Best	1256.160089	1256.140485	1256.112702
Mean	1256.162935	1256.146686	1256.115118
Variance	0.000004	0.000025	0.000002

In Figures 4-7, the black lines, blue lines and red lines represent the average trends of 10 runs for k-means, GA-k-means and MC, respectively. As can be seen from the figures, there is a fluctuation situation in each trend of MC, meaning the process of searching for the optimal object (cluster centers) at the high temperature. At a high temperature, the bad objects have high acceptance probabilities. It helps to avoid falling into local optimal solution. The acceptance probabilities of the bad objects decrease as the temperature reduces and the trends become stable. It can be seen that the convergence speed of MC is slower than that of k-means and GA-k-means. However, the optimal solution obtained by MC is more superior to the other two algorithms.

5. **Conclusions.** This paper presented a novel partition-based clustering algorithm based on tissue-like P systems. We designed a tissue-like P system that consisted of q elementary

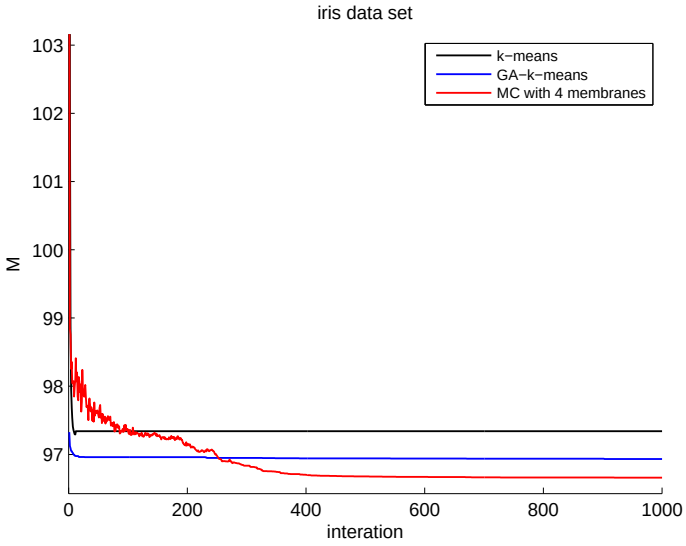

FIGURE 4. The average trend for *Iris*

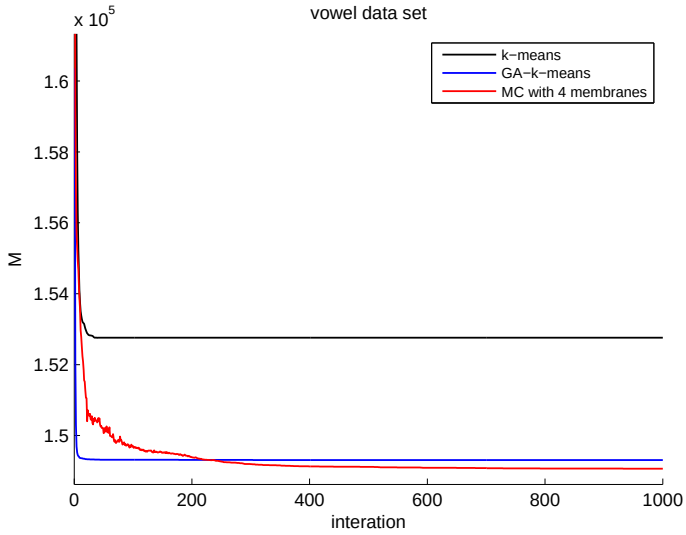

FIGURE 5. The average trend for *Vowel*

membranes to exploit the optimal cluster centers. There were an object, which represented a group of candidate cluster centers, and two types of evolution rules (simulated annealing rules and mutation rules) in each elementary membrane. The simulated annealing rules were inspired from the principle of the known simulated annealing algorithm and an adaptive modification strategy was utilized to reduce the sensitivity of initial temperature. The communication rules between the elementary membranes and between the elementary membranes and environment were used to control the sharing of the best objects.

The presented clustering algorithm based on tissue-like P systems was evaluated over two real-life data sets (*Iris* and *Vowel*) and two artificial data sets, and was compared with classical k-means algorithm and GA-based k-means algorithm. The comparison results indicated that the presented clustering algorithm was superior to other two algorithms in the aspect of clustering quality and had a good stability. However, the presented clustering algorithm had a slower convergence speed than k-means algorithm and GA-based k-means algorithm due to the use of simulated annealing mechanism. In our further

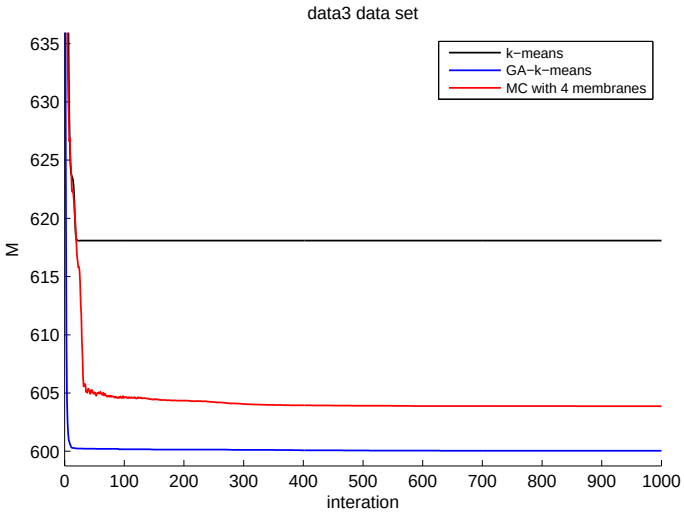

FIGURE 6. The average trend for *Data*3

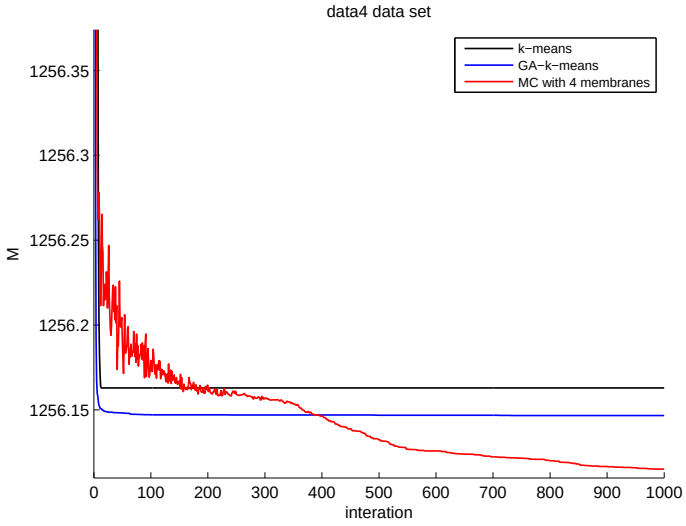

FIGURE 7. The average trend for *Data*4

works, other evolutionary strategies will be considered to overcome the weakness of the results.

Acknowledgement. This work was partially supported by the National Natural Science Foundation of China (No. 61170030), Research Fund of Sichuan Key Technology Research and Development Program (No. 2012GZ0019, No. 2013GZX0155), Open Research Funds of Key Laboratory of High Performance Scientific Computing (No. SZJJ2012-002) and Intelligent Network Information Processing (No. SZJJ2012-030), Chunhui Project Foundation of the Education Department of China (No. Z2012025, No. Z2012031), Importance Project Foundation of the Education Department of Sichuan province (No. 12ZA163), Innovation Fund of Postgraduate of Xihua University (No. YCJJ201317), and Importance Project Foundation of Xihua University (No. Z1122632), China.

REFERENCES

[1] Gh. Păun, Computing with membranes, *Journal of Computer System Sciences*, vol.61, no.1, pp.108-143, 2000.

[2] Gh. Păun, G. Rozenberg and A. Salomaa, *The Oxford Handbook of Membrance Computing*, Oxford University Press, New York, 2010.
[3] M. Ionescu, Gh. Păun and T. Yokomori, Spiking neural P systems, *Fundameta Informaticae*, vol.71, no.2-3, pp.279-308, 2006.
[4] J. Wang, L. Zhou, H. Peng and G. Zhang, An extended spiking neural P system for fuzzy knowledge representation, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(A), pp.3709-3724, 2011.
[5] H. Peng, J. Wang, M. J. Pérez-Jiménez, H. Wang, J. Shao and T. Wang, Fuzzy reasoning spiking neural P system for fault diagnosis, *Information Sciences*, vol.235, pp.106-116, 2013.
[6] J. Wang, P. Shi, H. Peng, M. J. Pérez-Jiménez and T. Wang, Weighted fuzzy spiking neural P systems, *IEEE Trans. on Fuzzy Systems*, vol.21, no.2, pp.209-220, 2013.
[7] R. Freund, Gh. Păun and M. J. Pérez-Jiménez, Tissue-like P systems with channel-states, *Theoretical Computer Science*, vol.330, no.1, pp.101-116, 2005.
[8] Gh. Păun and M. J. Pérez-Jiménez, Membrane computing: Brief introduction, recent results and applications, *BioSystem*, vol.85, no.1, pp.11-22, 2006.
[9] T. Y. Nishida, An application of P-system: A new algorithm for NP-complete optimization problems, *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, vol.5, pp.109-112, 2004.
[10] H. Wang, H. Peng, J. Shao and T. Wang, A thresholding method based on P systems for image segmentation, *ICIC Express Letters*, vol.6, no.1, pp.221-227, 2012.
[11] H. Peng, J. Wang, M. J. Pérez-Jiménez and P. Shi, A novel image thresholding method based on membrane computing and fuzzy entropy, *Journal of Intelligent & Fuzzy Systems*, vol.24, no.2, pp.229-237, 2013.
[12] J. Wang and H. Peng, Adaptive fuzzy spiking neural P systems for fuzzy inference and learning, *International Journal of Computer Mathematics*, vol.90, no.4, pp.857-868, 2013.
[13] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Elsevier Publication, Morgan Kaufmann, 2006.
[14] U. Maulik and S. Bandyopadhyay, Genetic algorithm based clustering technique, *Pattern Recognition*, vol.33, no.9, pp.1455-1465, 2000.
[15] T. Kanungo, D. M. Mount and N. S. Netanyahu, An efficient k-means clustering algorithm: Analysis and implementation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, no.7, pp.881-892, 2002.
[16] S. Bandyopadhyay and S. Saha, GAPS: A clustering method using a new point symmetry-based distance measure, *Pattern Recognition*, vol.40, pp.3430-3451, 2007.
[17] M. Laszlo and S. Mukherjee, A genetic algorithm that exchanges neighboring centers for k-means clustering, *Pattern Recognition Letters*, vol.28, pp.2359-2366, 2007.
[18] S. Kirkpatrick, C. Gelatt and M. Vecchi, Optimization by simulated annealing, *Science*, vol.22, pp.671-680, 1983.
[19] S. Bandyopadhyay, Simulated annealing using a reversible jump Markov chain monte carlo algorithm for fuzzy clustering, *IEEE Trans. on Knowledge and Data Engineering*, vol.17, no.4, pp.479-490, 2005.
[20] R. A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Human Genetics*, vol.7, no.2, pp.179-188, 1936.
[21] S. K. Pal and D. D. Majumder, Fuzzy sets and decision making approaches in vowel and speaker recognition, *IEEE Trans. on Systems, Man Cybernet*, vol.7, no.8, pp.625-629, 1977.