INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 759-771, October 2018.

A Spectral Clustering Algorithm Improved by P Systems

G. Chen, J. Hu, H. Peng, J. Wang, X. Huang

Guangchun Chen, Juan Hu, Hong Peng*,
Xiangnian Huang
School of Computer and Software Engineering
Xihua University, Chengdu, 610039, Sichuan, China
gchen@mail.xhu.edu.cn, jhu@mail.xhu.edu.cn
*Corresponding author: ph.xhu@hotmail.com
xhuang@mail.xhu.edu.cn

Jun Wang
School of Electrical and Information Engineering
Xihua University, Chengdu, 610039, Sichuan, China
wj.xhu@hotmail.com

Abstract: Using spectral clustering algorithm is difficult to find the clusters in the cases that dataset has a large difference in density and its clustering effect depends on the selection of initial centers. To overcome the shortcomings, we propose a novel spectral clustering algorithm based on membrane computing framework, called MSC algorithm, whose idea is to use membrane clustering algorithm to realize the clustering component in spectral clustering. A tissue-like P system is used as its computing framework, where each object in cells denotes a set of cluster centers and velocity-location model is used as the evolution rules. Under the control of evolution-communication mechanism, the tissue-like P system can obtain a good clustering partition for each dataset. The proposed spectral clustering algorithm is evaluated on three artificial datasets and ten UCI datasets, and it is further compared with classical spectral clustering algorithms. The comparison results demonstrate the advantage of the proposed spectral clustering algorithm.
Keywords: machine learning, spectral clustering, membrane computing, tissue-like P systems.

1 Introduction

Membrane computing initiated by Gheorghe Păun [17], was inspired from the structure and functioning of living cell as well as from the cooperation of cells in tissues, organs, and biological neural networks. Membrane computing is a class of distributed parallel computing models, known as P systems or membrane systems. In the past years, many variants of P systems have been proposed [7, 8, 11, 15, 16, 18, 19, 32, 41, 46], and they have been applied to different real-world problems [47], for example, combinatorial optimization [42, 44, 48], robots [1], image processing [4, 5, 22, 25, 40, 43], signal processing [23, 36, 45], knowledge representation [26, 34, 37], fault diagnosis [21, 28, 33, 35, 38, 39], ecology and system biology [3, 9, 10]. Most of membrane systems have been proved to be powerful (equivalent with Turing machine) and effective (able to solve the NP hard problems in a feasible time).

In the recent years, P systems were used to deal with data clustering problems. Zhao et al [49] presented an improved clustering algorithm, in which the rules in cell-like P systems were used to realize classical k-medoids algorithm. In Peng [24], evolution-communication P systems are used to deal with fuzzy clustering problems. In [27] and [29], two different mechanisms of P systems were considered to investigate automatic clustering problems. Liu et al [12] used a cell-like P systems with promoters and inhibitors to develop a k-medoids clustering algorithm.

In [20], fuzzy clustering problems were viewed as a multiobjective optimization problem and a tissue-like P system was designed to solve the optimization problem.

Spectral clustering is a popular method for solving clustering problems in a wide range of non-Euclidian spaces, linearly non-separable clusters and detecting non-convex patterns [13]. The key idea in spectral clustering is to achieve graph partitioning by performing eigen-decomposition of a graph Laplacian matrix. The obtained eigenvectors are used as the low dimensional representation of the data, and then the k-means algorithm is applied to generate the clusters. Spectral clustering approaches differ in how they define and construct the Laplacian matrix and thus which eigenvectors are selected to represent the partitioning. Moreover, different objective functions are used to derive the best cut. Chan et al. [2] proposed the ratio cut to minimize the total cost of the edges crossing the cluster boundaries, normalized by the size of the k clusters, to encourage balanced cluster sizes. Shi and Malik [31] established the normalized cut (NCut), which can measure the dissimilarities among groups and within clusters. In [6], Ding et al. proposed min-max cut criterion, which can avoid to segment the smaller subgraphs that contains only a few vertices. According to different partitioning criteria and spectral mapping methods, many different methods have been developed to realize spectral clustering algorithms. Perona and Freeman [30] proposed PF algorithm based on iterative spectrum, which is the simplest spectral clustering algorithm. Ng et al. [14] proposed the NJW algorithm, which is based on the K channel segmentation. However, there are a number of shortcomings in spectral clustering algorithms, for example, it is difficult to find the clusters with a large difference in density and their clustering effect depends on the selection of initial centers.

This paper focuses on application of membrane computing model in spectral clustering to overcome the shortcomings and presents a novel spectral clustering algorithm based on a membrane computing model, called MSC algorithm. A tissue-like P system is considered as a computing framework, and a membrane clustering algorithm is developed based on the computing framework and is embedded in a classical spectral clustering algorithm. To the best of our knowledge, this is the first attempt to use membrane computing model for improving spectral clustering algorithm.

The remainder of this paper is organized as follows. Section 2 reviews classical spectral clustering algorithm. Section 3 describes in detail the proposed membrane spectral clustering (MSC) algorithm. Experimental results and analysis are provided in Section 4. Conclusions is given in Section 5.

2 Spectral clustering and the NJW method

Spectral clustering method is a widely used graph-based approach for data clustering. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ in $R^{n\times d}$ with k clusters. We expect the dataset X will be transformed into a weighted undirected graph $G = (V, E)$, in which $V = \{x_i\}_{i=1}^n$ is the vertex set composed of n data points, and $E = \{w_{ij}\}_{i,j=1}^n$ is the set of weighted edges, where w_{ij} indicates the pairwise similarity between the x_i and x_j. V and E contain all vertices and edges, respectively, Let $W = (w_{i,j})_{1\le i,j\le n}$ be the affinity matrix. Usually, w_{ij} in affinity matrix can be measured by a Gaussian function:

$$w_{ij} = \begin{cases} e^{-\frac{d(x_i,x_j)^2}{\sigma^2}}, & i \neq j \\ 0, & i = j \end{cases} \tag{1}$$

The degree matrix D is a diagonal matrix, whose element D_{ii} is the degree of data point x_i, i.e., $D_{ii} = \sum_{j=1}^n w_{ij}$. Based on the two matrices, we can obtain the Laplacian matrix, L. There are three forms of Laplacian matrices: (i) unnormalized Laplacian matrix ($L = D - W$), and two

normalized Laplacian matrices, (ii) symmetric Laplacian matrix ($L_{sym} = D^{-1/2}WD^{-1/2}$) and (iii) random-walk Laplacian matrix ($L_{rw} = I - D^{-1}W$).

As a spectral approach for graph partitioning problem, NJW method is one of the most widely used spectral clustering algorithms. Its idea is to find a new representation of patterns on the first k eigenvectors of the Laplacian matrix. Algorithm 1 gives the details of NJW method.

Algorithm 1 NJW method

Input: $X \in R^{n\times d},\ k \in N$
Output: $V = \{v_i | i = 1, 2, \ldots, k\}$

1: Construct the affinity matrix $W \in R^{n\times n}$ according to Eq. (16);
2: Compute the degree matrix D;
3: Compute the normalized Laplacian matrix $L_{sym} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$;
4: Let $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k$ be the k least eigenvalues of L_{sym} and $p_1, p_2, \ldots, p_k$ be the corresponding eigenvectors; Construct the matrix $P = [p_1, p_2, \ldots, p_k] \in R^{n\times k}$, where p_i is ith column vector;
5: Construct the matrix Y from P by renormalizing each rows of P, i.e., $Y_{ij} = P_{ij}/(\sum_j P_{ij}^2)^{1/2}$;
6: Treat each row of Y as a point in R^k, and cluster them into k clusters $c_1, c_2, \ldots, c_k$ via k-means algorithm;
7: Output the clusters that corresponds to the original data set, $v_1, v_2, \ldots, v_k$, where $v_i = \{x_j | y_j \in c_i\}$.

3 Spectral clustering algorithm based on membrane computing framework

In this paper, we will try to use membrane computing algorithm (MCA) to replace the k-means component in classical spectral clustering algorithm to realize the optimal data partitioning. The spectral clustering algorithm optimized by membrane computing model is called MSC algorithm in this paper. By contrast, classical spectral clustering algorithm is called CSC algorithm. Figure 5 shows the structural comparison of CSC and MSC algorithms. From Figure 5, we can find that first component of MSC algorithm is the same to that of CSC algorithm, but MSC algorithm uses MCA algorithm rather than k-means algorithm in component 2. Therefore, in the following, we only describe the MCA algorithm. Since the core of MCA algorithm is a tissue-like P system, we first describe the tissue-like P system, and then illustrate the proposed MCA algorithm.

3.1 A tissue-like P system

We design a tissue-like P system (consisting of q cells) as the computing framework of MCA algorithm:

$$\Pi = (O_1, O_2, \ldots, O_q, R_1, R_2, \ldots, R_q, R', i_o)$$

where O_i is the set of objects in ith cell, R_i is the set of evolution rules in ith cell, R' is the set of communication rules between the cells, and the $i_o = 0$ indicates that the environment is the output region of the system.

Figure 2 shows the tissue-like P system, which consists of q cells labeled by $1, 2, \ldots, q$ respectively. Each cell has m objects, and the environment is labeled by 0. Denote by Z_{ij} the jth

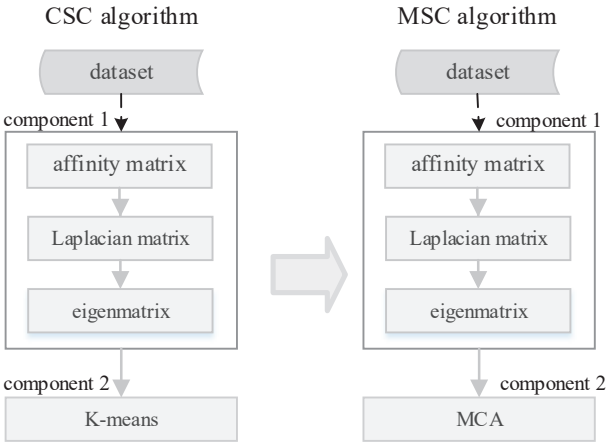


Figure 1: Structural comparison of CSC and MSC algorithms.

object in ith cell, $i = 1, 2, \ldots, q$, $j = 1, 2, \ldots, m$. The arrows in the figure indicate the communication of objects. The communication of objects is between these cells and the environment. The environment is also the output region of the system. When the system halts, the object in the environment is the optimal solution (a set of optimal cluster centers).

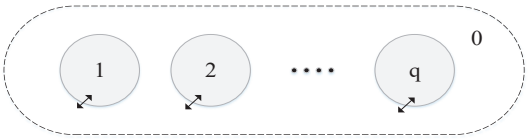


Figure 2: The designed tissue-like P system.

Each object in the cells, Z, is used to represent a set of candidate k cluster centers, $z_1, z_2, \ldots, z_k \in R^d$. Thus, object Z can be represented as a $k \times d$ dimensional vector:

$$Z = (z_{11}, z_{12}, \ldots, z_{1d}, \ldots, z_{i1}, z_{i2}, \ldots, z_{id}, \ldots, z_{k1}, z_{k2}, \ldots, z_{kd}) \tag{2}$$

where $(z_{i1}, z_{i2}, \ldots, z_{id})$ corresponds to ith candidate cluster center, $i = 1, 2, \ldots, k$.

These objects in cells will be evolved during the computation. Initially, a set of objects is generated randomly. Based on data points in a data set, we can determine a lower bound and an upper bound for each dimension, $A_j = \min\{x_{1j}, x_{2j}, \ldots, x_{nj}\}$, $B_j = \max\{x_{1j}, x_{2j}, \ldots, x_{nj}\}$, $j = 1, 2, \ldots, d$. Thus, $z_{ij} = rand([A_j, B_j])$, where $rand()$ denotes a random function that can generate the random number in $[A_j, B_j]$.

The tissue-like P system uses the communication rule of the form $< i, a; \lambda, 0 >$ to update the object in the environment, which means that object a in cell i is transported to environment 0, where λ denotes the empty object. The object in the environment is called the global optimal object, denoted by Z_{best}. For each cell, communication rule is used to communicate its best object to the environment and update the optimal object. The updating formula can be given as follows:

$$Z_{best} = \begin{cases} Z_{i,best}, & \text{if } J(Z_{i,best}) < J(Z_{best}) \\ Z_{best}, & \text{otherwise} \end{cases} \tag{3}$$

where $Z_{i,best}$ is the best object in ith cell. The object judgement is based on the following fitness function:

$$J(z_1, z_2, \ldots, z_k) = \sum_{i=1}^{k} \sum_{j=1}^{n} (u_{ij})^m \parallel x_j - z_i \parallel^2 \tag{4}$$

where u_{ij} denotes membership degree of x_j belonging to ith class, and m is a power exponent.

During the computation, tissue-like P system uses evolution rules to evolve the objects in cells. In this work, the velocity-location model of PSO is used as the evolution rules. The velocity-location model can be described as follows:

$$\begin{cases} V_j^i = wZ_j^i + c_1r_1(P_j^i - Z_j^i) + c_2r_2(Z_{best} - Z_j^i) \\ Z_j'^i = Z_j^i + V_j^i \end{cases} \tag{5}$$

where V_j^i corresponds to the speed of Z_j^i, $Z_j'^i$ is the new value of Z_j^i after evolving, and P_j^i is the best position so far for jth object in ith cell; w is the inertia weight constant, c_1 and c_2 are learning rate constants, and r_1 and r_2 are two random real numbers in $[0, 1]$. In the implementation of MCA algorithm, a linear decreasing strategy is used, i.e., $w = (0.9 - \frac{t}{2T})$, where t is the current iteration number and T is the maximum number of iterations.

In this paper, the maximum iteration number is used as halting condition. After the system halts, the best object Z_{best} in the environment is regarded as the solution. Finally, according to the optimal cluster centers, $c_1, c_2, \ldots, c_k$, N data points are classifies into k clusters.

3.2 Membrane clustering algorithm

As stated above, MCA algorithm is used as second component of the MSC algorithm. The MCA algorithm uses the designed tissue-like P system to automatically search for the optimal cluster centers for a data set to be clustered. Under the control of the evolution and communication rules, the P system continuously evolves the objects in cells and updates the global optimal object in the environment until the system halts. Figure 3 shows the flow chart of MCA algorithm.

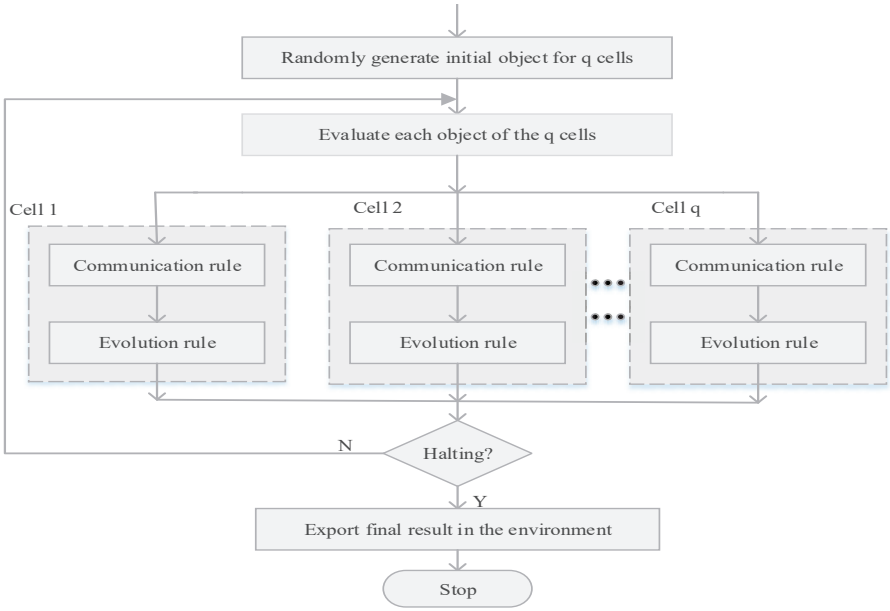
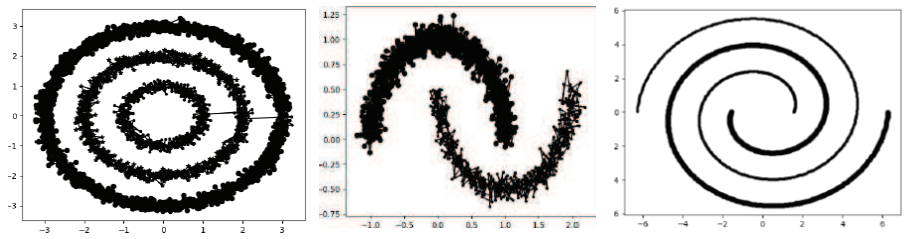


Figure 3: The flow chart of membrane clustering algorithm (MCA) used in MSC algorithm

4 Experimental results and analysis

4.1 Dataset

In order to evaluate the performance of MSC algorithm, three benchmark synthetic dataset and ten UCI dataset were used in experiments. The three synthetic dataset are *Threecircles*, *Twommoons* and *Spirl* respectively, shown in Figure 4(a)-(c). Table 1 gives the basic information for all data sets.



(a) (b) (c)

Figure 4: Three synthetic datasets. (a) Threecircle; (b) Twommoons; (c) Spirl.

Table 1: Data sets used in experiments

Datasets	No.of data	Dimension	No.of clusters
ThreeCircle	3603	2	3
Twommoons	1502	2	2
Spirl	944	2	2
Iris	150	4	3
wine	178	13	3
sonar	208	60	2
diabetes	1151	19	2
glass	214	9	6
ecoli	336	7	8
Heart	297	13	2
liver	345	6	2
ionosphere	351	34	2
sym	350	2	3

4.2 Experimental results

Three commonly used indexes of quality were used to measure the clustering performance.

(1) Adjusted Rand Index (ARI): $\rho_{ARI} \in [-1, 1]$.

This index measures the agreement between two compared partitions, namely, the ground truth (denoted as U) and the estimated by the tested clustering approach (denoted as V), and it is expressed by

$$\rho_{ARI} = \frac{a_{11} - (a_{11} + a_{01})(a_{11} + a_{10})/a_{00}}{(a_{11} + a_{01}) + (a_{11} + a_{10})/2 - (a_{11} + a_{01})(a_{11} + a_{10})/a_{00}} \tag{6}$$

where $a_{11} \in N$ is the number of sample pairs belonging to the same subset in U and in V, a_{10} is the number of sample pairs belonging to the same subset in U and to different subsets in V, $a_{01} \in N$ is the number of sample pairs belonging to different subset in U and

to the same one in V, and $a_{00} \in N$ is the number of sample pairs belonging to different subsets in U and in V.

(2) Purity Index (PUR): $\rho_{PUR} \in [0, 1]$.
This index matches the clustering partition V with the ground truth U as a weighted sum of the maximal precision values for each subset.

$$\rho_{PUR} = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |v_i \cap u_j| \tag{7}$$

(3) Jaccard Index (JAC): $\rho_{JAC} \in [0, 1]$.
This index matches the similarity among two sets, U and V, as follows:

$$\rho_{JAC} = \frac{a_{11}}{a_{11} + a_{10} + a_{01}} \tag{8}$$

In the experiment, two classical spectral clustering algorithms, K-SC and ϵ-SC, were introduced to implement two MSC algorithms, where membrane clustering algorithm is used to replace k-means component in the original spectral clustering algorithms. Thus, two MSC algorithms and the corresponding classical spectral clustering algorithms were compared in experiment. Table 2 and Table 3 show the comparison results of these algorithms on synthetic and UCI datasets, respectively. For each dataset, these tables provide the experimental results of four algorithms in terms of three indexes. Note that these experimental results are average value of 10 times independently running for each algorithm on a dataset. Moreover, we also provide the averages of these algorithms for each clustering index, respectively.

From table 2, we can see that the average value of MSC algorithm is the largest and can reach 1. The results show that the spectral clustering algorithm based on membrane computing framework has an obvious advantage in improving the average performance of spectral clustering algorithm. Comparison results of MSC and TSC on the UCI datasets show that K-SC and ϵ-SC algorithms based on membrane computing framework can significantly improve the Jaccard index, indicating that the proposed MSC algorithm is more robust and has a certain ability to deal with noise data. For the ARI and the Purity indexes, MSC algorithm achieves a comparable result in comparison to the classical algorithm.

Iris dataset is used as an example to analyze the influences of parameters in MSC algorithm. Figure 5 (a)-(c) shows the influences of three parameters, including bandwidth ϵ of the Gaussian kernel function, the number of cells m and the maximum number of iterations $Maxstep$. As can be seen from the figures, MSC algorithm is more sensitive to m and ϵ, and the curve of parameter $Maxstep$ rises slowly and finally tends to straight line, which indicates that the performance of the algorithm is not improved when the maximum number of iterations is reached.

Table 2: Clustering quality assessment results (synthetic datasets).

Dataset	Quality measure	K-SC		ϵ-SC	
		k-means	MCA	k-means	MCA
Threecircle	ARI	1.0	1.0	0.51	1.0
	Purity	1.0	1.0	0.84	1.0
	Jaccard	1.0	1.0	0.56	1.0
Twommoons	ARI	0.37	1.0	0.59	1.0
	Purity	0.81	1.0	0.89	1.0
	Jaccard	0.56	1.0	0.70	1.0
Spirl	ARI	1.0	1.0	0.89	1.0
	Purity	1.0	1.0	0.73	1.0
	Jaccard	1.0	1.0	0.95	1.0
Average	ARI	0.79	1.0	0.66	1.0
	Purity	0.93	1.0	0.82	1.0
	Jaccard	0.85	1.0	0.73	1.0

Table 3: Clustering quality assessment results (UCI repository datasets).

Dataset	Quality measure	K-SC		ϵ-SC	
		k-means	MCA	k-means	MCA
Iris	ARI	0.44	0.66	0.63	0.75
	Purity	0.87	0.86	0.84	0.90
	Jaccard	0.50	0.64	0.60	0.71
wine	ARI	0.03	0.30	0.29	0.0
	Purity	0.57	0.67	0.53	0.40
	Jaccard	0.24	0.37	0.38	0.34
Sonar	ARI	0.02	-0.01	0.00	-0.0058
	Purity	0.57	0.51	0.55	0.50
	Jaccard	0.34	0.48	0.34	0.48
diabetes	ARI	0.16	0.01	0.16	0.0
	Purity	0.70	0.53	0.70	0.53
	Jaccard	0.43	0.51	0.43	0.50
glass	ARI	0.17	0	0.16	0.01
	Purity	0.51	0.36	0.52	0.36
	Jaccard	0.25	0.26	0.33	0.26
ecoli	ARI	0.37	0.20	0.48	0.42
	Purity	0.56	0.45	0.66	0.70
	Jaccard	0.32	0.29	0.42	0.65
Heart	ARI	0.31	0.04	0.37	0.40
	Purity	0.78	0.55	0.80	0.75
	Jaccard	0.49	0.51	0.52	0.63
liver	ARI	0	-0.02	-0.01	0.12
	Purity	0.65	0.58	0.98	0.67
	Jaccard	0.36	0.51	0.50	0.56
ionosphere	ARI	0.15	0.01	0.13	0.25
	Purity	0.70	0.64	0.68	0.73
	Jaccard	0.44	0.54	0.41	0.55
sym	ARI	0.56	0.75	0.49	0.38
	Purity	0.68	0.87	0.75	0.54
	Jaccard	0.79	0.72	0.43	0.43
Average	ARI	0.28	0.20	0.28	0.23
	Purity	0.66	0.61	0.70	0.61
	Jaccard	0.41	0.49	0.43	0.52

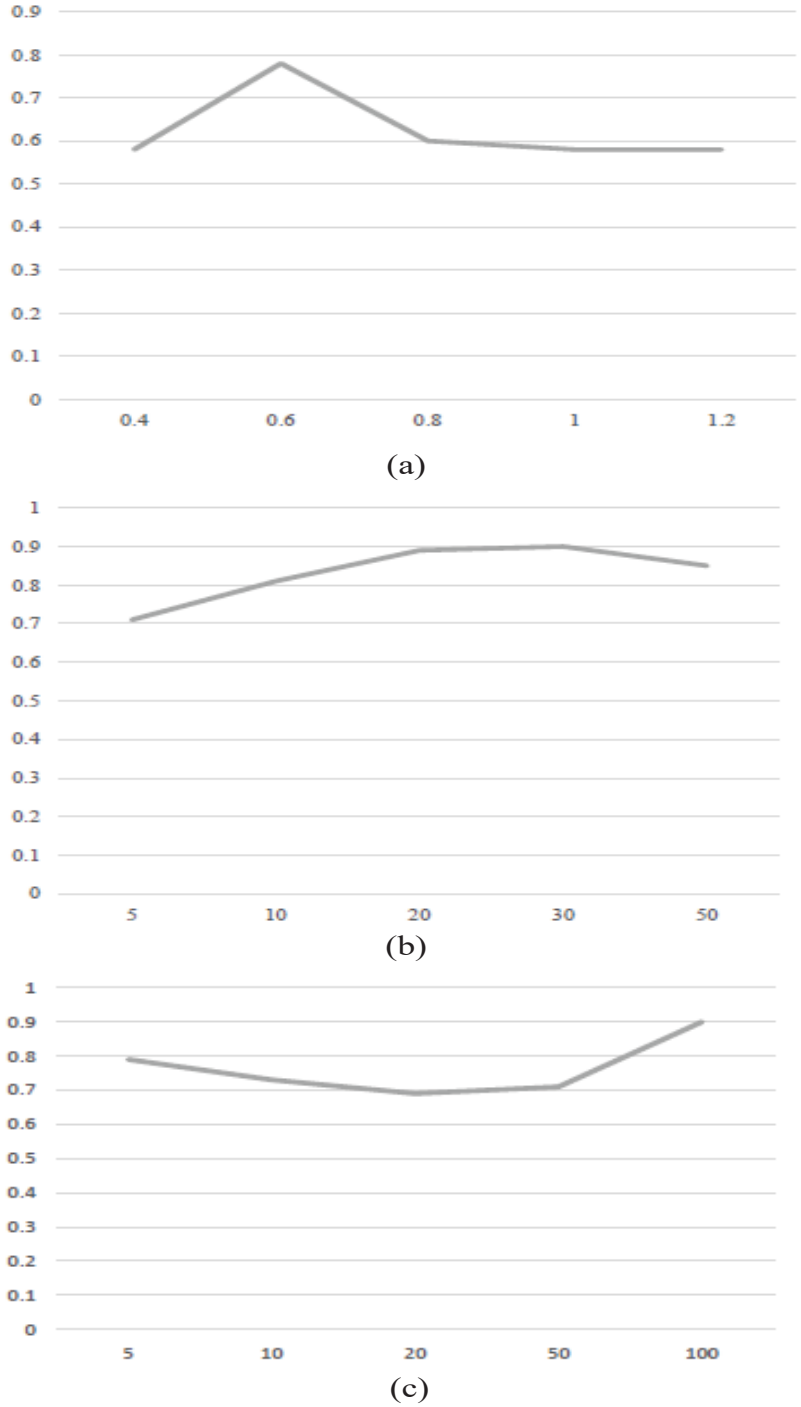


Figure 5: Free parameter analysis over UCI datasets based on the Purity: (a) bandwidth ϵ of the Gaussian kernel function; (b) the number of cells m; (c) the maximum number of iterations $Maxstep$.

5 Conclusion

In this paper, we used membrane computing framework to develop a novel spectral clustering algorithm, called MSC algorithm. The core component of MSC algorithm is a tissue-like P system which is composed of several cells and uses the improved PSO algorithm as evolution mechanism. We evaluated the performance of the proposed algorithm on three artificial data sets and ten UCI datasets. The results show that compared with the classical spectral clustering algorithm, the proposed algorithm can improve the clustering performance. This study also demonstrates the effectiveness of using the membrane computing framework to solve data clustering problems.

MSC algorithm used membrane clustering algorithm (MCA) instead of k-means component in classical spectral clustering algorithm, which searches for the optimal solution by both the evolution of objects in multiple cells and the communication of objects between the cells. It is well known that membrane computing is a distributed computing model. However, MSC algorithm is not implemented in parallel due to limitation of the computer's serial architecture. Therefore, our further work is to discuss the parallel implementation of MSC algorithm on GPGPU and/or FPGA.

Funding

This work was partially supported by the National Natural Science Foundation of China (No. 61472328), Chunhui Project Foundation of the Education Department of China (Nos. Z2016143 and Z2016148), Research Foundation of the Education Department of Sichuan province (No. 17TD0034), China, and the Innovation Fund of Postgraduate, Xihua University(no. ycjj2017073).

Bibliography

[1] Buiu, C.; Vasile, C.; Arsene, O. (2012); Development of membrane controllers for mobile robots, *Information Sciences*, 187, 33-51, 2012.

[2] Chan, P.K.; Schlag, M.D.F.; Zien, J.Y. (1993); Spectral k-way ratio-cut partitioning and clustering, *DAC*, 749-754, 1993.

[3] Colomer, A.M.; Margalida, A.; Pérez-Jiménez, M.J. (2013); Population dynamics P system (PDP) models: a standarized protocol for describing and applying novel bio-inspired computing tools, *Plos One*, 4, 1-13, 2013.

[4] Díaz-Pernil, D.; Berciano, A.; Peña-Cantillana, F.; Gutiérrez-Naranjo, M.A. (2013); Segmenting images with gradient-based edge detection using membrane computing, *Pattern Recognition Letters*, 34(8), 846-855, 2013.

[5] Díaz-Pernil, D.; Peña-Cantillana, F.; Gutiérrez-Naranjo, M.A. (2013); A parallel algorithm for skeletonizing images by using spiking neural P systems, *Neurocomputing*, 115, 81-91, 2013.

[6] Ding, C.; He, X.; Zha, H.; Gu, M.; Simon, H. (2001); Spectral min-max cut for graph partitioning and data clustering, *Technical Report TR-2001-XX*, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.

[7] Dzitac, I. (2015); Impact of membrane computing and P systems in ISI WoS. celebrating the 65th birthday of Gheorghe Păun, *International Journal of Computers Communications & Control*, 10(5), 617-626, 2015.

[8] Freund, R.; Păun, G.; Pérez-Jiménez, M.J. (2005); Tissue-like P systems with channel-states, *Theoretical Computer Science*, 330, 101-116, 2005.

[9] Garcia-Quismondo, M.; Levin, M.; Lobo-Fernández, D. (2017); Modeling regenerative processes with Membrane Computing, *Information Sciences*, 381, 229-249, 2017.

[10] Gheorghe, M.; Manca, V.; Romero-Campero, F.J. (2010); Deterministic and stochastic P systems for modelling cellular processes, *Natural Computing*, 9(2), 457-473, 2010.

[11] Ionescu, M.; Păun G.; Yokomori, T. (2006); Spiking neural P systems, *Fundamenta Informaticae*, 71, 279-308, 2006.

[12] Liu, X.; Zhao, Y.; Sun, W. (2016); K-medoids-based consensus clustering based on cell-like P systems with promoters and inhibitors, *Bio-inspired Computing - Theories and Applications*, 95-108, 2016.

[13] Luxburg, U.V. (2007); A tutorial on spectral clustering, *Statistics and Computing*, 17(4), 395-416, 2007.

[14] Ng, A.Y., Jordan, M., Weiss, Y. (2001); On spectral clustering: analysis and an algorithm, *Proc Nips*, 849-856, 2001.

[15] Pan, L.; Wang, J.; Hoogeboom, H.J. (2012); Spiking neural P systems with astrocytes, *Neural Computation*, 24(3), 805-825, 2012.

[16] Pan, L.; Păun, G. (2009); Spiking neural p systems with anti-spikes, *International Journal of Computers Communications & Control*, 4(3), 273-282, 2009.

[17] Păun, G. (2000); Computing with membranes, *Journal of Computer System Sciences*, 61(1), 108-143, 2000.

[18] Păun, G.; Rozenberg, G.; Salomaa, A. (2010); *The Oxford Handbook of Membrance Computing*, Oxford Unversity Press, New York, 2010.

[19] Păun, G. (2016); Membrane computing and economics: a general view, *International Journal of Computers Communications & Control*, 11(1), 105-112, 2016.

[20] Peng, H.; Shi, P.; Wang, J.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. (2017); Multiobjective fuzzy clustering approach based on tissue-like membrane systems, *Knowledge-Based Systems*, 125, 74-82, 2017.

[21] Peng, H.; Wang, J.; Ming, J.; Shi, P.; Pérez-Jiménez, M.J.; Yu, W.; Tao, C. (2018); Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems, *IEEE Transaction on Smart Grid*, 2018. Available at http://dx.doi.org/10.1109/TSG.2017.2670602.

[22] Peng, H.; Wang, J.; Pérez-Jiménez, M.J. (2015); Optimal multi-level thresholding with membrane computing, *Digital Signal Processing*, 37, 53-64, 2015.

[23] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2014); The framework of P systems applied to solve optimal watermarking problem, *Signal Processing*, 101, 256-265, 2014.

[24] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2015); An unsupervised learning algorithm for membrane computing, *Information Sciences*, 304(20), 80-91, 2015.

[25] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Shi, P. (2013); A novel image thresholding method based on membrane computing and fuzzy entropy, *Journal of Intelligent and Fuzzy Systems*, 24(2), 229-237, 2013.

[26] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Wang, H.; Shao, J.; Wang, T. (2013); Fuzzy reasoning spiking neural P system for fault diagnosis, *Information Sciences*, 235(20), 106-116, 2013.

[27] Peng, H.; Wang, J.; Shi, P.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2016); An extended membrane system with active membrane to solve automatic fuzzy clustering problems, *International Journal of Neural Systems*, 26, 1-17, 2016.

[28] Peng, H.; Wang, J.; Shi, P.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2017); Fault diagnosis of power systems using fuzzy tissue-like P systems, *Integrated Computer-Aided Engineering*, 24, 401-411, 2017.

[29] Peng, H.; Wang, J.; Shi, P.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. (2015); An automatic clustering algorithm inspired by membrane computing, *Pattern Recognition Letters*, 68(15), 34-40, 2015.

[30] Perona, P.; Freeman, W. (1998); A factorization approach to grouping, *Computer Vision ECCV'98*, Springer, 655-670, 1998.

[31] Shi, J.; Malik, J. (2000); Normalized cuts and image segmentation, IEEE Transactions on pattern analysis and machine intelligence, 22(8), 888-905, 2000.

[32] Song, T.; Pan, L., Păun, G. (2014), Spiking neural P systems with rules on synapses, *Theoretical Computer Science*, 529, 82-95, 2014.

[33] Tu, M.; Wang, J.; Peng, H.; Shi, P. (2014); Application of adaptive fuzzy spiking neural P systems in fault diagnosis of power systems, *Chin. Jour. Elect.*, 23(1), 87-92, 2014.

[34] Wang, J.; Peng, H. (2013); Adaptive fuzzy spiking neural P systems for fuzzy inference and learning, *International Journal of Computer Mathematics*, 90(4), 857-868, 2013.

[35] Wang, J.; Peng, H.; Tu, M.; Pérez-Jiménez, M.J. (2016); A fault diagnosis method of power systems based on an improved adaptive fuzzy spiking neural P systems and PSO algorithms, *Chin. Jour. Elect.*, 25(2), 320-327, 2016.

[36] Wang, J.; Shi, P.; Peng, H. (2016); Membrane computing model for IIR filter design, *Information Sciences*, 329, 164-176, 2016.

[37] Wang, J.; Shi, P.; Peng, H.; Pérez-Jiménez, M.J.; Wang, T. (2013); Weighted fuzzy spiking neural P system, *IEEE Trans. Fuzzy Syst.*, 21(2), 209-220, 2013.

[38] Wang, T.; Zhang, G.X.; Zhao, J.B.; He, Z.Y.; Wang, J., Pérez-Jiménez, M.J. (2015); Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems, *IEEE Trans. Power Syst.*, 30(3), 1182-1194, 2015.

[39] Xiong, G.; Shi, D.; Zhu, L.; Duan, X. (2013); A new approach to fault diagnosis of power systems using fuzzy reasoning spiking neural P systems, *Mathematical Problems in Engineering*, 2013(1), 211-244, 2013.

[40] Yahya, R.I.; Hasan, S.; George, L.E.; Alsalibi, B. (2015); Membrane computing for 2D image segmentation, *International Journal of Advances in Soft Computing and its Application*, 7(1), 35-50, 2015.

[41] Zeng, X.; Zhang, X.; Song, T.; Pan, L. (2014); Spiking neural P systems with thresholds, *Neural Computation*, 26(7), 1340-1361, 2014.

[42] Zhang, G.; Cheng, J.; Gheorghe, M.; Meng, Q. (2013); A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, *Applied Soft Computing*, 13(3), 1528-1542, 2013.

[43] Zhang, G.; Gheorghe, M.; Li, Y. (2012); A membrane algorithm with quantum-inspired subalgorithms and its application to image processing, *Natural Computing*, 11(4), 701-717, 2012.

[44] Zhang, G.; Gheorghe, M.; Pan, L.; Pérez-Jiménez, M.J. (2014); Evolutionary membrane computing: a comprehensive survey and new results, *Information Sciences*, 279, 528-551, 2014.

[45] Zhang, G.; Liu, C.; Rong, H. (2010); Analyzing radar emitter signals with membrane algorithms, *Mathematical and Computer Modelling*, 52, 1997-2010, 2010.

[46] Zhang, X.; Pan, L.; Păun, A. (2015); On the universality of axon P systems, *IEEE Transactions on Neural Networks and Learning Systems*, 26(11), 2816-2829, 2015.

[47] Zhang, G.; Pérez-Jiménez, M.J.; Gheorghe, M. (2017); *Real-life Applications With Membrane Computing*, Springer, 2017.

[48] Zhang, G.; Rong, H.; Neri, F.; Pérez-Jiménez, M.J. (2014); An optimization spiking neural P system for approximately solving combinatorial optimization problems, *International Journal of Neural Systems*, 24, 1-16, 2014.

[49] Zhao, Y.; Liu, X.; Qu, J. (2012); The k-medoids clustering algorithm by a class of P system, *Journal of Information & Computational Science*, 9(18), 5777-5790, 2012.