ROCK clustering algorithm based on the P system with active membranes

YUZHEN ZHAO, XIYU LIU, WENPING WANG
School of Management Science and Engineering
Shandong Normal University
Jinan, Shandong
CHINA
zhaoyuzhen_happy@126.com, sdxyliu@163.com, 64759428@qq.com

Abstract: - The ROCK algorithm plays an important role in data mining and data analysis, which can help people discover knowledge from large amounts of data. In this paper, an improved ROCK algorithm based on the P system with active membranes is constructed. Since the P system has great parallelism, it could reduce the computational time complexity and is suitable for the clustering problem. All the rules of the proposed algorithm are designed in this paper. Experimental results show that the proposed algorithm is appropriate for clustering large dataset. The proposed improved ROCK algorithm is a new attempt in applications of membrane system and it provides a novel perspective of cluster analysis.

Key-Words: - Clustering Algorithm; Hierarchical Clustering; ROCK Algorithm; Membrane Computing; P System; Membrane System

1 Introduction

The implicit pattern or knowledge can be extracted from huge amounts of data by data mining technology. Cluster analysis, as one of the important part of data mining, is an effective method to pack data and discover useful information from these data. It is a hot area of research in machine learning, statistics, biology, data mining and many other fields. Cluster partitions the whole data into several clusters with the intra-cluster data similar and the inter-cluster data dissimilar. After data are packed by cluster, these data are analysed and useful information can be discovered. For instance, the dense and sparse areas can be recognized, the global distribution patterns and the interesting relationship between data attributes can be discovered [1]. ROCK algorithm is a typical hierarchical clustering algorithm which can produce a more meaningful data dealing with the data sets with Boolean attributes or categorical attributes. This algorithm adopts a more global view by considering the neighborhood of point pairs to cluster and it has shown the powerful features when dealing with some real life data sets [2]. With emergence of the big data, which shows several characteristics, such as volume, variety, velocity, and value, the data mining is paid a greater attention [3]. People need more efficient algorithms to deal with the big data. Membrane computing, as a new biological calculation model, has big parallelism and can significantly improve the efficiency of algorithms.

The ROCK clustering algorithm, which is used to solve the typical clustering problem to cluster N transactions into k clusters, measures the similarity between clusters through the goodness measure. The goodness measure, which uses the similarity between points and the neighborhood links information at the same time, considers the similarity between clusters from a more global view， so the ROCK algorithm can obtain better clustering results [2]. Dutta et al. proposed a new algorithm QROCK based on ROCK which computes the clusters by determining the connected components of the graph [4]. Gao et al. proposed a collaborative algorithm based on ROCK algorithm to recommend books [5]. Tyagi et al. used the ROCK algorithm to optimize the query searching time [6].

P system, the computing model of the new biological calculation method membrane computing, is abstracted based on the structure and function of the cell. P system has the great parallelism so it can decrease the time complexity of computing [7]. The researchers have proposed three kinds of P systems: Cell-like P Systems, Tissue-like P Systems and Neural-like P Systems and have studied the computational ability and the computational efficiency [8-10]. Recently, the new variant P systems (numerical P systems [11], enzymatic numerical P systems [12], spatial P systems [13] and so on) were proposed. For providing a concrete model of P systems, Verlan proposed the formal

framework for P systems and Gheorghe and Ipate proposed the kernel P systerms[14, 15]. Gazdag used P Systems with Active Membranes to solve the SAT problem [16]. Cienciala et al. used 2D P colonies to model the surface runoff [17]. Banu-Demergian et al. used the geometric membrane structure to realize the finite interactive systems scenarios [18]. Sun and Liu used the P system with active membranes to solve the density-based clustering [19].

Since the volume characteristic of big data, the existing calculation model is very difficult to gain the calculation results in a short period of time. The parallelism of P system makes it has the higher computing power, so it reduces the time complexity of data processing and satisfies the requirement of improving the processing speed of the big data [8, 9]. Membrane computing has been widely applied to many fields like combinatorial problem, finite state problems and graph theory. But the application in data mining, especially in cluster analysis, is less. Using the membrane computing to the clustering can reduce the time complexity of clustering, so it has a certain theoretical and practical significance.

The ROCK algorithm and the membrane computing are combined in this paper to solve the typical clustering problem of clustering N transactions into k clusters. Firstly, the membrane structure is initialized. Secondly, the rules of algorithm are determined according to the thought of the traditional ROCK algorithm. Lastly, the experiment samples are mapped to the objects in P system and clustering of experiment samples is achieved by the execution of the rules. Two experiments are designed to verify the feasibility of the proposed algorithm.

This paper has been organized as follows. Section 2 presents the ROCK algorithm based on the P system with active membranes and lists all the rules. In Section 3, an illustrative experiment and the mushroom data set of UCI dataset are described. Then the experimental results are presented. Finally, some conclusions are included in Section 4.

2 Methods

The main idea of proposing the ROCK clustering algorithm based on P system is that the ROCK algorithm is a typical hierarchical clustering algorithm which has shown the powerful features when dealing with some real life data sets and the big parallelism of P system can decrease the time complexity of ROCK algorithm to meet people's requirements on the computing efficiency. The flow chart of proposed algorithm is shown in Fig. 1.

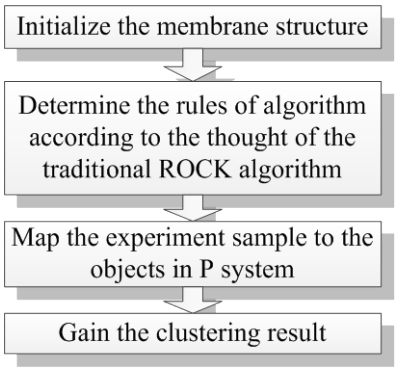

Fig. 1 The flow chart of proposed algorithm

2.1 The ROCK Algorithm [2]

For the data set containing the Boolean or categorical attributes data, the traditional clustering algorithm uses the distance function. However, experiments show that the distance function can't produce high quality clusters. The ROCK algorithm, a kind of hierarchical clustering algorithm, deals with these data sets through a more global view by considering the neighborhood of the pair of points. If two points having a similar neighborhood, they may belong to the same cluster. ROCK algorithm has shown the powerful features when dealing with some real life data sets.

The three notions which realize the advantages of ROCK algorithm are introduced.

Definition 1: Neighbors The neighbors of a point are the points which are similar to it. In the database with categorical attributes, the database consists of transactions. Each transaction consists of some items and each item has a number of fixed values. The similarity between two transactions is set as the numbers of the same value items. If the numbers are larger or equal to a certain threshold, the two transactions are called neighbors.

Definition 2: Links $links(p_i, p_j)$ is the number of the common neighbors between transaction p_i and transaction p_j. The larger the $links(p_i, p_j)$, the more neighbors they have.

Definition 3: Goodness Measure The goodness measure is the measure which is used to find the most similar pair of clusters to be merged at each step by measuring the similarity between clusters.

$Links[C_i, C_j]$ is used to defined the number of links between cluster i and cluster j. That is to say, $link[C_i, C_j] = \sum_{p_q \in C_i, p_r \in C_j} links(p_q, p_r)$. Then the goodness measure for merging the clusters i and j is defined as follow:

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{n_i + n_j} \quad (1)$$

This goodness measure is equivalent to the average number of links of transactions in these two clusters. The pair of clusters with the maximum goodness measure is the best pair of clusters to be merged.

Here, suppose the dataset has n transactions and each transaction has m items. They are expected to be clustered to *k* clusters.

Firstly, each transaction is seen as a cluster. Then, the two clusters with the largest goodness measure are merged in each step until *k* clusters are left.

In some real life dataset, the ROCK clustering algorithm has shown its powerful function. Compared with the traditional hierarchical clustering algorithm, it can produce more meaningful clustering results.

2.2 P Systems with Active Membranes

P system is abstracted from the process of cells managing chemical substances as a new membrane computing model. Based on different biochemical reactions in tissues or biological cells, three kinds of models are proposed: Cell-like P Systems, Tissue-like P Systems and Neural-like P Systems. Cell-like P System imitates the function and structure of the cells, and it includes the membrane structure, rules and objects as basic elements. The basic membrane structure is shown in Fig.2. Membranes divide the whole system into different regions. The skin membrane is the outermost membrane (In Fig.2, membrane 1 is the skin membrane). A membrane is a basic membrane if there are no membranes in it (In Fig.2, membrane 2, 3, 5, 8, 9 and 7 are basic membranes) and a membrane is a non-elemental membrane otherwise (In Fig.2, membrane 1, 4, and 6 are non-elemental membranes). Rules and objects exist in regions. Usually the objects are indicated by strings or characters. Rules are used to process objects or membranes in corresponding region. The rules are executed uncertainly and maximum concurrently. P System can be divided into three types from the angle of kinds of rules: transition P system, P system with communication rules and P system with active membranes [8].

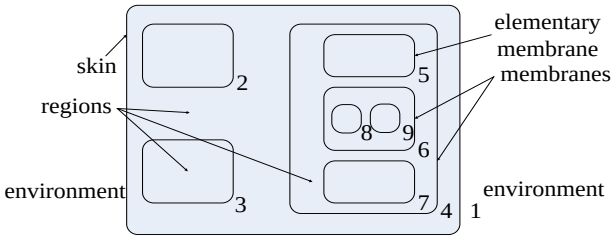

Fig.2 The basic membrane structure

P systems with active membranes change the membrane structure when executing rules. So space of exponential growth can be generated in linear operation steps. This is very helpful to solve the computationally hard problems within feasible time.

In general, a P system with active membranes of degree m is a construct:

$$\prod = (O, \mu, w_1, w_2, ..., w_m, R_1, R_2, ..., R_m, i_0) \quad (2)$$

Where:

1. O is an alphabet. Elements in it are called objects;
2. μ is a membrane structure, each membrane has its label, $H=\{1,2,\ldots,m\}$ is the label set ;
3. $w_i(i=1,2,\ldots,m)$ is the objects in membrane i ;
4. $R_i(i=1,2,\ldots,m)$ is the rules of membrane i .

The basic rule is in the form of $(u \to v)_r$ [9]. u is a string composed of objects in O and v is a string in the form of $v=v'$ or $v=v'\ \delta$. v' is a string over $\{a_{here}, a_{out}, a_{in_j} \mid a \in O, 1 \le j \le m\}$. δ is a symbol not in O. It means after executing the rule this membrane will be dissolved. r is the promoters or the inhibitors and it is in the form of $r=r'$ or $r = \neg r'$ .A rule can execute only when the promoters r' appear and a rule can stop only when the inhibitors appear. The radius of this rule $u \to v$ is the length of u. R_i is the set of the rules in region i. ρ_i is the precedence relation which defines the partial order relation over R_i .High priority rule is executed prior[7]. The active membranes have rules as follows:

(a) object evolution rules:

$$[_h\, a \to v]_h^e, h \in H, e \in \{+, -, 0\}, a \in O, v \in O^*$$

Object a evolves into v when the electric charge of membrane $h^{\#}$ is e. The electric charge of membrane $h^{\#}$ can be ignored if it is 0 which means that membrane $h^{\#}$ has no polarity.

(b) communication rules:

$$a[_h]_h^{e_1} \to [_h b]_h^{e_2}, [_h a]_h^{e_1} \to [_h]_h^{e_2} b,$$
$$h \in H, e_1, e_2 \in \{+, -, 0\}, a, b \in O$$

Object a out of membrane h changes to object b and enters the membrane or object a in membrane h

changes to object *b* and comes out of the membrane when the electric charge of membrane $h^{\#}$ is e_1. And the polarization of the membrane can be changed, but the label of membrane cannot.

(c) dissolving rules:

$[_h a]_h^e \to b \;\; h \in H, e_1, e_2 \in \{+,-,0\}, a, b \in O$

Membrane h will be dissolved with all objects in membrane *h* be kept except object *a* which is changed to object *b* when electric charge in membrane $h^{\#}$ is e.

(d) Division rules:

$[_h a]_h^{e_1} \to [_h b]_h^{e_2} [_h c]_h^{e_3}$

$h \in H, e_1, e_2, e_3 \in \{+,-,0\}, a, b, c \in O$

Membrane *h* can be separated into two membranes *h* with all objects duplicated except object *a* which is changed to object *b* and *c* respectively when the electric charge in membrane $h^{\#}$ is e_1. These two new membranes may have different polarizations.

(e) Fusion rules:

$[_h b]_h^{e_2} [_h c]_h^{e_3} \to [_h a]_h^{e_1}$

$h \in H, e_1, e_2, e_3 \in \{+,-,0\}, a, b, c \in O$

This rule is opposite to the division rules shown above.

5. i_0 is the output membrane which is used to save the final calculation result[11].

The rules are used maximum parallel and uncertainly in each membrane when calculating. The P system will halt after some steps if no more rules can be executed and these objects in output membrane is the final result. The P system will not halt if rules are always executed, then this calculation is invalid, and there is no result being outputted.

2.3 A P System with Active Membranes based on ROCK Clustering Algorithm

2.3.1 Construction of the P system with Active Membranes based on ROCK Clustering Algorithm

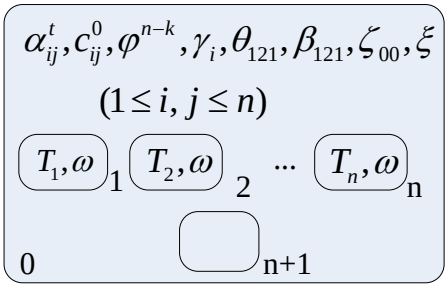

Fig.3 The P system based on the ROCK clustering method

In Fig.3, the structure of the P system based on ROCK clustering algorithm proposed in this paper is depicted. The point a_{ij}^t represents the value of the j_{th} item in the i_{th} transaction is t. The specific algorithm is followed:

The algorithm is defined as follows:

$$\prod = (O, \mu, M_0, M_1, ..., M_n, M_{n+1}, R_0, R_1, ..., R_n, R_{n+1}, \rho) \quad (3)$$

Where:

1) $O = \{T_1, T_2, ..., T_n, \omega, \xi, \alpha_{ij}^t, c_{ij}^0, \varphi^{n-k}, \theta, \theta', \delta_{ij}, d_{ij}^t, \tau_{ij}, \eta, \gamma_i, \theta_{ijt}, \beta_{ijp}, \zeta_{ij}\} (1 \le i, j, p \le n, 1 \le t \le m+1)$

O represents the collection of objects in the P system.

2) $\mu = [_0 [_1]_1 [_2]_2 ... [_n]_n [_{n+1}]_{n+1}]_0$

μ represents the membrane structure of the P system.

3) $M_0 = \{\alpha_{ij}^t, c_{ij}^0, \varphi^{n-k}, \gamma_i, \theta_{121}, \beta_{121}, \zeta_{00}, \xi (1 \le i, j \le n)\}$

$M_1 = \{T_1, \omega\}$

$M_2 = \{T_2, \omega\}$

...

$M_n = \{T_n, \omega\}$

$M_{n+1} = \{\lambda\}$

M represents the collection of initial objects in each membrane. M_{n+1} is the output membrane of this P system.

The rules in R_0:

$$r_1 = \{\theta_{ijt} a_{it}^p a_{jt}^p \to \theta_{ij(t+1)} a_{it}^p a_{jt}^p \delta_{ij}\} \cup \{(\theta_{ijt})_{\neg a_{it}^p a_{jt}^p} \to \theta_{ij(t+1)} \mid 1 \le i, j \le n, 1 \le t \le m\} \cup \{\theta_{ij(m+1)} \to \theta_{i(j+1)1} \mid 1 \le i, j \le n\} \cup \{\theta_{in(m+1)} \to \theta_{(i+1)(i+2)1} \mid 1 \le i \le n\}$$

$$r_2 = \{\delta_{ij} \alpha_{ij} \to \lambda \mid 1 \le i < j \le n\}$$

$$r_3 = \{\delta_{ij}^t \to w_{ij} w_{ji} \mid 1 \le i < j \le n, t > 0\}$$

$$r_4 = \{\beta_{ijt} c_{ij}^p w_{it} w_{jt} \to \beta_{ij(t+1)} c_{ij}^{p+1} w_{it} w_{jt}\} \cup \{(\beta_{ijt})_{\neg w_{it} w_{jt}} \to \beta_{ij(t+1)} \mid 1 \le i, j, t \le n, 0 \le p \le n\} \cup \{\beta_{ij(n+1)} \to \beta_{i(j+1)1} \mid 1 \le i, j \le n\} \cup \{\beta_{in(n+1)} \to \theta_{(i+1)(i+2)1} \mid 1 \le i \le n\}$$

$$r_5 = \{\xi c_{i_1 j_1}^{p_1} c_{i_2 j_2}^{p_2} ... c_{i_t j_t}^{p_t} \to c_{i_1 j_1}^{p_1} c_{i_2 j_2}^{p_2} ... c_{i_t j_t}^{p_t} d'^{p_1}_{i_1 j_1} d'^{p_2}_{i_2 j_2} ... d'^{p_t}_{i_t j_t} d_{i_1 j_1}^0 d_{i_2 j_2}^0 ... d_{i_t j_t}^0\} \quad (1 \le i, j \le n, p > 0)$$

$$r_6 = \{(d'^{q_i + q_j}_{ij} d_{ij}^p)_{r_i^{q_i} r_j^{q_j}} \to d_{ij}^{p+1} \mid 1 \le i, j, p \le n\}$$

$$r_7 = \{d'^{p}_{ij} \to e'^{2p}_{ij} \mid 1 \le i, j, p \le n\}$$

$$r_8 = \{(e'^{q_i+q_j}_{ij}\, e'^{p}_{ij}\, d^{q}_{ij})_{r_i^{q_i} r_j^{q_j}} \to d^{q+1}_{ij}\}$$

$$(1 \le i, j \le n, 0 \le p, q \le n)$$

$$r_9 = \{e'^{t}_{ij} \to \lambda \mid 1 \le i, j \le n, 0 < t\}$$

$$r_{10} = \{d^{0}_{ij}\zeta_{pt} \to \zeta_{ij} \mid 1 \le i, j \le n, 0 \le p, t \le n\}$$

$$r_{11} = \{d^{p_1}_{i_1 j_1} d^{p_2}_{i_2 j_2} ... d^{p_t}_{i_t j_t} \to d^{p_1-1}_{i_1 j_1} d^{p_2-1}_{i_2 j_2} ... d^{p_t-1}_{i_t j_t}\}$$

$$(1 \le i, j \le n, p_q > 0, 1 \le q \le t)$$

$$r_{12} = \{[[\omega]_i [\omega]_j \zeta_{ij} \gamma_j^p]_0 \to [[\omega]_i \zeta_{ij} \gamma_i^p]_0 \mid 1 \le i, j, p \le n\}$$

$$r_{13} = \{\zeta_{ij} c_{ij}^p \to \zeta_{ij} \mid 1 \le i, j \le n\}$$

$$r_{14} = \{\varphi\zeta_{ij} \to \zeta_{00}\xi\}$$
$$\bigcup \{(c_{jt}^{p_1})_{\neg c_{it}^{p2} \cap \neg c_{ti}^{p3}} \to c_{it}^{p}\}$$
$$\bigcup \{(c_{tj}^{p_1})_{\neg c_{it}^{p2} \cap \neg c_{ti}^{p3}} \to c_{ti}^{p}\}$$
$$\bigcup \{c_{jt}^{p} c_{it}^{p'} \to c_{it}^{p+p'}\}$$
$$\bigcup \{c_{jt}^{p} c_{ti}^{p'} \to c_{ti}^{p+p'}\}$$
$$\bigcup \{c_{tj}^{p} c_{it}^{p'} \to c_{it}^{p+p'}\}$$
$$\bigcup \{c_{tj}^{p} c_{ti}^{p'} \to c_{ti}^{p+p'}\}$$
$$\bigcup \{(\zeta_{ij})_{\neg\varphi} \to \delta_{1_{in\ 1,2,..n}}\}$$
$$(1 \le i, j, t \le n, p \ge 0, q > k)$$

The rules in $R_i (1 \leqslant i \leqslant n)$:

$$r_1 = \{(\delta_i \omega)_{T_i} \to \delta_{i+1} \omega T_i\}$$
$$\bigcup \{(\delta_i)_{\neg T_i} \to \delta_{i+1}\} (1 \le i \le n)$$

$$r_2 = \{\delta_{n+1}\omega \to \omega_{in\ (n+1)}\}$$

2.3.2 Outline of Rules

In this membrane system shown in Fig.3, most computational processes are taken place in the skin membrane 0.

In the initial configuration, each transaction is assumed as a cluster and all the clusters are represented by membranes which are named from 1 to n. In these membranes, object T_i represents the i_{th} transaction in the database.

The computation begins when objects a_{ij}^{t}, which represents that the value of the j_{th} item in the i_{th} transaction is t, are put into membrane 0.

The object δ_{ij} is generated when the t_{th} items of transaction i and j are the same. After the whole transaction database is traversed, the number of object δ_{ij} is equal to the number of the same value items between the corresponding transaction i and transaction j. This is also the similarity between the two transactions. The number of the objects α_{ij} is the threshold of the similarity. When the number of α_{ij} is less than the number of objects δ_{ij}, the transactions i and j are similarly enough and the objects w_{ij} and w_{ji} are generated to show the two transactions are neighbors.

There is a link between transaction i and transaction j, when objects w_{it} and w_{jt} exist at the same time. After the t traversed from 1 to n, all the links between transaction i and transaction j are obtained, represented by object c_{ij}^{p}.

The goodness measure of cluster i and cluster j is computed by formula (1). This function is realized by rules r_6~r_{10}. The number of object γ_i represents the number of transactions in cluster i. Each object c_{ij}^{p} generated p objects d'_{ij} and d_{ij}^{0}. If there are q_i r_i and q_j r_j, each q_{i+} q_j objects d'_{ij} made the superscript of object d_{ij}^{p} adds 1. This process will continue until there are not enough objects d'_{ij}. Then the objects d'_{ij} are doubled. If the number of objects d'_{ij} is more than or equal to q_{i+} q_j, the superscript of object d_{ij}^{p} adds 1. If not, let the remaining objects d'_{ij} disappear. These steps realize the function of rounding. Let all the superscript of objects d_{ij}^{p} decrease at the same time. When the superscript of one object d_{ij}^{p} is 0, the object ζ_{pt} is changed to ζ_{ij}. That is to say, the subscript of object ζ_{ij} is used to show the two clusters with the biggest goodness measure at the end. Clusters i and j are merged into one cluster i. The objects γ_j are changed to object γ_i to show the number of transactions in cluster j is added to cluster i. The objects c_{ij}^{p} are got rid because there is no cluster j any more. If there are p links between cluster i and cluster t and p' links between cluster j and cluster t, and the p' links are added to the corresponding cluster i and cluster t so the object c_{it}^{p} is changed to $c_{it}^{p+p'}$. Then one object ξ is generated to repeat the cluster process until only k clusters are left.

Then the objects δ_1 are put into are put into these k clusters. In each membrane, all the transactions T_i are arranged in a string ω and put into the output membrane (n+1).

3 Experiments and Analysis

3.1 Illustrative Experiment

The following experiment illustrates how the membrane system shown in Fig.3 works.

There are eight customers buying something from the seven kinds of goods in a supermarket. The numbers from 1 to 7 represent the goods and the c_i represents the i_{th} customer ($i \in [1,8]$). The detail

information about the goods bought by consumer is given in Table 1.

Table 1 the transactional data of one supermarket

customers	goods
c_1	1,2,3
c_2	1,2,4
c_3	1,2,5
c_4	1,3,4
c_5	1,2,6
c_6	1,2,7
c_7	1,6,7
c_8	2,6,7

The eight costumers should be clustered into two clusters. If two transactions have two same items, denoted by the value, these transactions are defined as neighbors. The object a_{ij}^{1} is used for showing that the i_{th} customer bought the j_{th} good. Whereas the the object a_{ij}^{1} is not input when the i_{th} customer has not bought the j_{th} good.

The membrane system algorithm of this experiment is shown in Fig.4.

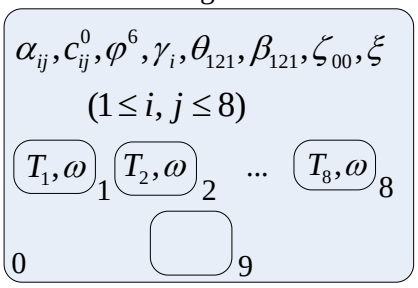

Fig.4 The P system is used to cluster eight transactions into two clusters

As the time change, the objects in each membrane of the system are shown in Table 2. Since there are a lot of steps and some steps are almost the same, only the parts of the first circulation of the clustering are listed. Only the objects in membranes 0, 5 and 6 are listed, because objects in other membranes have no changes in the first circulation. According to the proposed algorithm, cluster 5 and the cluster 6 are merged firstly. Cluster 5 and cluster 6 are merged firstly according to our algorithm.

In Fig.5, the different transactions are denoted by different nodes. If two transactions are neighbors, a connection edge is drawn between them. Two transactions are more similar if there are more links between them. Then they are more likely to belong to the same cluster. It can be seen from the Fig.5 that the 5_{th} transaction and the 6_{th} transaction are located nearly in the center of the figure and they have the most neighbors and links. This example verifies the feasibility of the proposed algorithm.

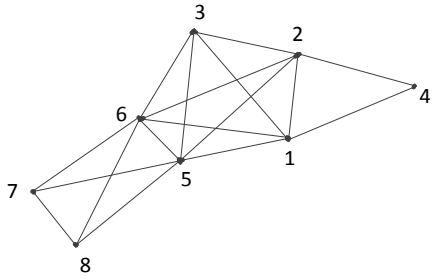

Fig.5 The link relationships between the eight transactions

Table 2 the objects change of the first clustering circulation

membrane number	0	5	6
t_0	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{121},\beta_{121},\zeta_{00},\xi$	T_5,ω	T_6,ω
t_1	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{122},\beta_{121},\zeta_{00},\xi,\delta_{12}(r_1)$	T_5,ω	T_6,ω
t_2	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{123},\beta_{121},\zeta_{00},\xi,\delta_{12}^{\ 2}(r_1)$	T_5,ω	T_6,ω
t_3	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{124},\beta_{121},\zeta_{00},\xi,\delta_{12}^{\ 2}(r_1)$	T_5,ω	T_6,ω
...			
t_{197}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{121},\zeta_{00},\xi,\delta_{12}^{\ 2},\delta_{13}^2,\delta_{14}^2,\delta_{15}^2,\delta_{16}^2,\delta_{17}^1,\delta_{18}^1,\delta_{23}^2,\delta_{24}^2,\delta_{25}^2,\delta_{26}^2,\delta_{27}^1,\delta_{28}^1,\delta_{34}^1,\delta_{35}^2,\delta_{36}^2,\delta_{37}^1,\delta_{38}^1,\delta_{45}^1,\delta_{46}^1,\delta_{47}^1,\delta_{56}^2,\delta_{57}^2,\delta_{58}^2,\delta_{67}^2,\delta_{68}^2,\delta_{78}^2,(r_1)$	T_5,ω	T_6,ω
t_{198}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{121},\zeta_{00},\xi,\delta_{12},\delta_{13},\delta_{14},\delta_{15},\delta_{16},\delta_{23},\delta_{24},\delta_{25},\delta_{26},\delta_{35},\delta_{36},\delta_{56},\delta_{57},\delta_{58},\delta_{67},\delta_{68},\delta_{78},(r_2)$	T_5,ω	T_6,ω
t_{199}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{121},\zeta_{00},\xi,w_{12},w_{13},w_{14},w_{15},w_{16},w_{21},w_{23},w_{24},w_{25},w_{26},w_{31},w_{32},w_{35},w_{36},w_{41},w_{42},w_{51},w_{52},w_{53},w_{56},w_{57},w_{58},w_{61},w_{62},w_{63},w_{65},w_{67},w_{68},w_{75},w_{76},w_{78},w_{85},w_{86},w_{87}(r_3)$	T_5,ω	T_6,ω
t_{200}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{122},\zeta_{00},\xi,w_{12},...,w_{87}(r_4)$	T_5,ω	T_6,ω
t_{201}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{123},\zeta_{00},\xi,w_{12},...,w_{87}(r_4)$	T_5,ω	T_6,ω

Table 2(continued) the objects change of the first clustering circulation

membrane number	0	5	6
t_{202}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{124},\zeta_{00},\xi,w_{12},...,w_{87},c_{12}^1(r_4)$	T_5,ω	T_6,ω
t_{203}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},c_{ij}^0,\varphi^6,\gamma_i,\theta_{788},\beta_{125},\zeta_{00},\xi,w_{12},..,w_{87},c_{12}^2(r_4)$	T_5,ω	T_6,ω
...			
t_{396}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},\xi,w_{12},...,w_{87},c_{12}^4,c_{13}^3,c_{14}^1,c_{15}^3,c_{16}^3,c_{17}^2,c_{18}^2,c_{23}^3,c_{24}^1,c_{25}^3,c_{26}^3,c_{27}^2,c_{28}^2,c_{34}^2,$ $c_{35}^3,c_{36}^3,c_{37}^2,c_{38}^2,c_{45}^2,c_{46}^2,c_{47}^0,c_{48}^0,c_{56}^5,c_{57}^2,c_{58}^2,c_{67}^2,c_{68}^2,c_{78}^2(r_4)$	T_5,ω	T_6,ω
t_{397}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d'^4_{12},d'^3_{13},d'^1_{14},d'^3_{15},d'^3_{16},d'^2_{17},d'^2_{18},d'^3_{23},d'^1_{24},$ $d'^3_{25},d'^3_{26},d'^2_{27},d'^2_{28},d'^2_{34},d'^3_{35},d'^3_{36},d'^2_{37},d'^2_{38},d'^2_{45},d'^2_{46},d'^0_{47},d'^0_{48},d'^5_{56},d'^2_{57},d'^2_{58},d'^2_{67},d'^2_{68},d'^2_{78},d_{12}^0,$ $d_{13}^0,d_{14}^0,d_{15}^0,d_{16}^0,d_{17}^0,d_{18}^0,d_{23}^0,d_{24}^0,d_{25}^0,d_{26}^0,d_{27}^0,d_{28}^0,d_{34}^0,d_{35}^0,d_{36}^0,d_{37}^0,d_{38}^0,d_{45}^0,d_{46}^0,d_{47}^0,d_{48}^0,d_{56}^0,d_{57}^0,d_{58}^0,$ $d_{67}^0,d_{68}^0,d_{78}^0(r_5)$	T_5,ω	T_6,ω
t_{398}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d'^1_{13},d'^1_{14},d'^1_{15},d'^1_{16},d'^1_{23},d'^1_{24},d'^1_{25},d'^1_{26},d'^1_{35},$ $d'^1_{36},d'^1_{56},d_{12}^2,d_{13}^1,d_{14}^0,d_{15}^1,d_{16}^1,d_{17}^1,d_{18}^1,d_{23}^1,d_{24}^0,d_{25}^1,d_{26}^1,d_{27}^1,d_{28}^1,d_{34}^1,d_{35}^1,d_{36}^1,d_{37}^1,d_{38}^1,d_{45}^1,d_{46}^1,d_{47}^0,d_{48}^0,$ $d_{56}^2,d_{57}^1,d_{58}^1,d_{67}^1,d_{68}^1,d_{78}^1(r_6)$	T_5,ω	T_6,ω
t_{399}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,e'^2_{13},e'^2_{14},e'^2_{15},e'^2_{16},e'^2_{23},e'^2_{24},e'^2_{25},e'^2_{26},e'^2_{35},e'^2_{36},$ $e'^2_{56},d_{12}^2,d_{13}^1,d_{14}^0,d_{15}^1,d_{16}^1,d_{17}^1,d_{18}^1,d_{23}^1,d_{24}^0,d_{25}^1,d_{26}^1,d_{27}^1,d_{28}^1,d_{34}^1,d_{35}^1,d_{36}^1,d_{37}^1,d_{38}^1,d_{45}^1,d_{46}^1,d_{47}^0,d_{48}^0,d_{56}^2,$ $d_{57}^1,d_{58}^1,d_{67}^1,d_{68}^1,d_{78}^1(r_7)$	T_5,ω	T_6,ω
t_{400}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},w_{12},...w_{87},c_{12}^4,...,c_{78}^2,d_{12}^2,d_{13}^2,d_{14}^1,d_{15}^2,d_{16}^2,d_{17}^1,d_{18}^1,d_{23}^2,d_{24}^1,d_{25}^2,d_{26}^2,$ $d_{27}^1,d_{28}^1,d_{34}^1,d_{35}^2,d_{36}^2,d_{37}^1,d_{38}^1,d_{45}^1,d_{46}^1,d_{47}^0,d_{48}^0,d_{56}^3,d_{57}^1,d_{58}^1,d_{67}^1,d_{68}^1,d_{78}^1(r_8)$	T_5,ω	T_6,ω
t_{401}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{47},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{12}^2,d_{13}^2,d_{14}^1,d_{15}^2,d_{16}^2,d_{17}^1,d_{18}^1,d_{23}^2,d_{24}^1,d_{25}^2,d_{26}^2,$ $d_{27}^1,d_{28}^1,d_{34}^1,d_{35}^2,d_{36}^2,d_{37}^1,d_{38}^1,d_{45}^1,d_{46}^1,d_{48}^0,d_{56}^3,d_{57}^1,d_{58}^1,d_{67}^1,d_{68}^1,d_{78}^1(r_{10})$	T_5,ω	T_6,ω
t_{402}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{48},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{12}^2,d_{13}^2,d_{14}^1,d_{15}^2,d_{16}^2,d_{17}^1,d_{18}^1,d_{23}^2,d_{24}^1,d_{25}^2,d_{26}^2,$ $d_{27}^1,d_{28}^1,d_{34}^1,d_{35}^2,d_{36}^2,d_{37}^1,d_{38}^1,d_{45}^1,d_{46}^1,d_{56}^3,d_{57}^1,d_{58}^1,d_{67}^1,d_{68}^1,d_{78}^1(r_{10})$	T_5,ω	T_6,ω
t_{403}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{48},w_{12},..,w_{87},c_{12}^4,...,c_{78}^2,d_{12}^1,d_{13}^1,d_{14}^0,d_{15}^1,d_{16}^1,d_{17}^0,d_{18}^0,d_{23}^1,d_{24}^0,d_{25}^1,d_{26}^1,$ $d_{27}^0,d_{28}^0,d_{34}^0,d_{35}^1,d_{36}^1,d_{37}^0,d_{38}^0,d_{45}^0,d_{46}^0,d_{56}^2,d_{57}^0,d_{58}^0,d_{67}^0,d_{68}^0,d_{78}^0(r_{11})$	T_5,ω	T_6,ω
t_{404}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{14},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{12}^1,d_{13}^1,d_{15}^1,d_{16}^1,d_{17}^0,d_{18}^0,d_{23}^1,d_{24}^0,d_{25}^1,d_{26}^1,d_{27}^0,$ $d_{28}^0,d_{34}^0,d_{35}^1,d_{36}^1,d_{37}^0,d_{38}^0,d_{45}^0,d_{46}^0,d_{56}^2,d_{57}^0,d_{58}^0,d_{67}^0,d_{68}^0,d_{78}^0(r_{10})$	T_5,ω	T_6,ω
...			
t_{418}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{78},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{12}^1,d_{13}^1,d_{15}^1,d_{16}^1,d_{23}^1,d_{25}^1,d_{26}^1,d_{35}^1,d_{36}^1,d_{56}^2(r_{10})$	T_5,ω	T_6,ω
t_{420}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{12},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{13}^0,d_{15}^0,d_{16}^0,d_{23}^0,d_{25}^0,d_{26}^0,d_{35}^0,d_{36}^0,d_{56}^1(r_{10})$	T_5,ω	T_6,ω
t_{428}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{36},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{56}^1(r_{10})$	T_5,ω	T_6,ω
t_{429}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{36},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,d_{56}^0(r_{11})$	T_5,ω	T_6,ω
t_{430}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{56},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2(r_{10})$	T_5,ω	T_6,ω
t_{431}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{56},w_{12},...,w_{87},c_{12}^4,...,c_{78}^2,\gamma_1,\gamma_2,\gamma_3,\gamma_4,\gamma_5{}^2,\gamma_7,\gamma_8(r_{12})$	T_5,T_6,ω	×
t_{432}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^6,\gamma_i,\theta_{788},\beta_{788},\zeta_{56},w_{12},...,w_{87},c_{12}^4,c_{13}^3,c_{14}^1,c_{15}^3,c_{16}^3,c_{17}^2,c_{18}^2,c_{23}^3,c_{24}^1,c_{25}^3,c_{26}^3,c_{27}^2,c_{28}^2,c_{34}^2,$ $c_{35}^3,c_{36}^3,c_{37}^2,c_{38}^2,c_{45}^2,c_{46}^2,c_{47}^0,c_{48}^0,c_{57}^2,c_{58}^2,c_{67}^2,c_{68}^2,c_{78}^2,\gamma_1,\gamma_2,\gamma_3,\gamma_4,\gamma_5{}^2,\gamma_7,\gamma_8(r_{13})$	T_5,T_6,ω	×
t_{433}	$a_{11}^1,...,a_{87}^1,\alpha_{ij},\varphi^5,\gamma_i,\theta_{788},\beta_{788},\zeta_{00},\xi,w_{12},...,w_{87},c_{12}^4,c_{13}^3,c_{14}^1,c_{15}^6,c_{17}^2,c_{18}^2,c_{23}^3,c_{24}^1,c_{25}^6,c_{27}^2,c_{28}^2,c_{34}^2,c_{35}^6,c_{37}^2,$ $c_{38}^2,c_{45}^4,c_{47}^0,c_{48}^0,c_{57}^2,c_{58}^4,c_{57}^2,c_{78}^2,\gamma_1,\gamma_2,\gamma_3,\gamma_4,\gamma_5{}^2,\gamma_7,\gamma_8(r_{14})$	T_5,T_6,ω	×

3.2 Applied Numerical Experiment and Discuss

The mushroom dataset of UCI database contains 8124 samples [20]. Each sample contains 22 attribute values. All the data are belong to two classes: the edible and the poisonous. In this paper, the first 1000 data constitute the experiment set.

The 1000 samples are numbered form 1 to 1000. Then the ROCK algorithm based on P system is applied into cluster them. In this experiment, two data, which have 18 or more same attribute values,

are regarded as neighbors. The 7 clusters obtained by the proposed algorithm are listed in Table 3.

As the result shown in the Table 2, the sizes of first cluster, the third cluster, the fourth cluster and the fifth cluster are almost the same (Their sizes are 102, 122, 85, and 80, respectively). With the 591 samples and the only 1 sample, the second cluster and the sixth cluster are the largest cluster and the smallest cluster, respectively, which shows huge differences between them.

All the clusters obtained by the ROCK algorithm are pure clusters. That is to say, in any cluster, all data are either all poisonous or all edible. The error rate is 0%, which indicated the algorithm proposed in this paper is suitable for applied into this dataset. Schlimmer also reviewed these 1000 instances by STAGGER and gained 5% error rate. The proposed algorithm gained the better clustering results [21].

Table 3 The 7 clusters got through the ROCK algorithm

class	The serial number of data belonging to the corresponding cluster
1	1,4,9,14,18~20,22,26,32,38,44,54,55,79,82, 115,121,123,136,139,181,186, 206,222,229,232,244,252,262,270,272,281,300, 312,316,328,331,358,381,386,400, 403,415,418,423,492,493, 506,524,533,535,543,557,566,569,580,591,594,596,599,600, 614,654,655,663,664,695,698,699, 701,726,733,734,738,749,777,786,789,795,796,799, 803,813~815,836,838,842,860, 907,909,910,928,933,943,950,951,957,967,985, 1000
2	2,3,6~8,10~13,21,23~25,27,28,31,33~35,40~42,45~53,56,58~60,62~65,67~69,72,74,75,78,80,85,87~89,92~94,96~99, 100,102~111,114,116,118,122,125,127,129~132,134,138,140~142,144,145,148~151,153,156~162,164~170,172,174~180,184,185,187,189,190,193,195,197~199, 201~203,205,207~212,215~219,221,223,225~228,230,233,235,237~243,246,248,249,251,254~256,258~264,268,269,271,273~278,280,282,284~289,292~297, 302,304,305,307~309,311,313,315,318,321,323~327,329,330,332~335,337,338,340~343,345,348,352~355,357,359,360,364,365,367,369,371~373,375,376,378~380,383,384,387,389~393,395,397,398, 401,402,404~411,413,414,416,419,421,422,424,425,427,428,431~434,436,438~449,451~453,455,457~464,466~472,478,480~482,486,487,489,490,494~496,498~499, 500~505,512,517~523,526~528,530~532,534,537,539,541,544,546,547,550,553,555,556,559~561,565,567,570,571,573~576,578,579,581~586,590,593,595,597,598, 602,604~606,608,609,615,616,618,619,621,622,624,626,629,630,632,634~638,640,~645,647~649,651~653,656,658,659,661,662,665~671,675,678~680,682,684,686,688~693,696,697, 703~707,709~711,713~719,721,722,725,727,729~731,735,736,739,742~748,751,752,754~757,759~763,765,769~773,776,778~781,783~785,788,790~794,797, 800~802,804~809,811,812,816~818,820,821,823~826,829,833~835,837,839,840,843,845,847~849,851,852,856,858,859,861~866,868,870~873,875~880,883~885,887~890,892~899, 900~901,905,911,913,916,917,923,930,934,937,941,948,952,956,960,977,979,982,990,991,997,999
3	5,15,17,57,66,81,84,86,95, 101,124,126,128,146,147,152,191, 204,250,267,279,298, 301,310,346,349,351,361~363,370,382,388,396, 412,420,429,456,497, 507,508,516,536,542,545,552,564,568,592, 601,603,607,611~613,617,631,639,657,673,694, 700,702,712,724,728,732,758,766,775,798, 846,854,882, 902~904,906,908,912,914,915,919~922,925~927,931,932,936,938,939,942,944,946,947,949,953~955,961,962,965,966,969~971,974,976,980,983,984,986~989,992,993,995,996
4	16,29,37,43,61,70,83,90,91, 112,117,120,143,154,163,171,182,183,188,196, 214,220,231,234,253,257,266,283,290,291, 339,350,356,366,374,377,394,399, 430,465,474,476,477,484,488, 510,511,514,515,525,529,538,548,551,577,587~589, 610,620,625,628,660,672,676,677,687, 708,720,740,741,764,768,782,787, 822,831,844,853,855,867,869, 929,935,994

Table 3(continued) The 7 clusters got through the ROCK algorithm

class	The serial number of data belonging to the corresponding cluster
5	30,36,39,71,73,76,77, 113,119,133,135,137,155,173,192,194, 200,213,224,236,245,247,265,299, 303,314,317,319,320,322,344,347,368, 417,426,435,437,454,473,475,479,483,485,491, 509,513,540,549,554,558,562,563,572, 623,627,646,650,674,681,683, 723,737,750,753,767,774, 810,819,827,828,830,832,841,850,857,881,886, 924,959,998
6	306
7	336,385, 450, 633,685, 874,891, 918,940,945,958,963,964,968,972,973,975,978,981

4 Conclusion

With the advent of the era of big data, the traditional way of data processing is more and more difficult to meet people's requirement to efficiency. Profit from the great parallelism, P system can decrease the time complexity of computing and improve the computational efficiency. Recent years, as a new biological computing method, the theory of membrane computing has been adequately studied. Due to its great parallelism, it has been applied into many fields like combinatorial problem, finite state problems and graph theory but not enough. A ROCK clustering algorithm based on the P system with active membranes is constructed in this paper. This proposed algorithm is suitable for cluster analysis by experiment. This paper applies membrane computing into the typical clustering algorithm ROCK algorithm enlarges the research field which applies membrane computing to solve clustering algorithm. But the accurately time complexity is not given. The following research work will focus on the theoretically analyze of the proposed algorithm's time complexity. Additionally, there are many clustering methods and membrane computing can be applied to a variety of other clustering methods.

5 Acknowledgment

This work was supported by the Natural Science Foundation of China(No.61170038), Natural Science Foundation of Shandong Province, China (No.ZR2011FM001), Humanities and Social Sciences Project of Ministry of Education, China (No.12YJA630152), Social Science Fund of Shandong Province, China (No.11CGLJ22), Science-Technology Program of the Higher Education Institutions of Shandong Province, China (No. J12LN65), Research Award Foundation for Outstanding Young Scientists of Shandong Province, China (No.BS2012DX041).

References:

1. J. Han and M. Kambr, *Data Mining Concepts and Techniques*. USA: Elsevier Inc., 2012.
2. S. Guha, R. Rastogi and K. Shim, Rock: A Robust Clustering Algorithm for Categorical Attributes, *Information System. J.*, Vol. 25, No. 5, pp. 345–366, 2000.
3. C. Dunren, M. Safran and Z. Y. Peng, From Big Data to Big Data Mining: Challenges, Issues, and Opportunities, *Database Systems for Advanced Applications*. Springer Berlin Heidelberg, 2013, pp. 1–15, January. 2013.
4. M. Dutta, A. K. Mahanta and A. K. Pujari. QROCK: A quick version of the ROCK algorithm for clustering of categorical data. *Pattern Recognition Letters*, 2005, 26(15): 2364-2373.
5. H. Gao, H. Wang and Y. Zhang, A Book Collaborative Filtering Recommendation Based on ROCK Clustering and Transitive Similarity. *Control and Automation Publication Group*, 2010, 26 (11-3).
6. A. Tyagi and S. Sharma. Implementation of ROCK Clustering Algorithm for the Optimization of Query Searching Time. *International Journal on Computer Science & Engineering*, 2012, 4(5).
7. G. Paun. A quick introduction to membrane computing, *The Journal of Logic and Algebraic Programming. J.*, Vol. 79, pp. 291–294, 2010.
8. R. Freund, M. Oswald and G. Paun. Catalytic and Purely Catalytic P system and P Automata: Control Mechnisms for Obtaining

Computational Completeness. *14th International Conference, CMC14*, pp.317-320, 2013.
9. R. Freund, L. Kari, M. Oswald and P. Sosik. Computationally universal P systems without priorities: two catalysts are sufficient. *Theoretical Computer Science. J*, Vol. 330, No. 2, pp. 251-266, 2005.
10. G. Paun, G. Rozenberg and A. Salomaa. *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.
11. G. Paun and R. Paun, Membrane computing and economics: Numerical P systems. *Fundamenta Informaticae. J*, Vol. 73, No. 1, pp.213-227, 2006.
12. A. Pavel, O. Arsene and C. Buiu. Enzymatic numerical P systems-a new class of membrane computing systems. *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on. IEEE*, pp.1331-1336, 2010.
13. R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, G. Pardini and L. Tesei. Spatial P systems. *Natural Computing. J*, Vol. 10, No. 1, pp.3-16, 2011.
14. S. Verlan. Using the Formal Framework for P Systems. *14th International Conference, CMC14*, pp.37-38, 2013.
15. M. Gheorghe and F. Ipate. Kernel P Systems - A Survey. *14th International Conference, CMC14*, pp.9-10, 2013.
16. Z. Gazdag. Solving SAT by P Systems with Active Membranes in Linear Time in the Number of Variables. *14th International Conference, CMC14*, pp. 167-180, 2013.
17. L. Cienciala, L. Ciencialova and M. Langer. Modelling of Surface Runoff using 2D P colonies. *14th International Conference, CMC14*, pp.81-94, 2013.
18. I.T. Banu-Demergian and G. Stefanescu. The Geometric Membrane Structure of Finite Interactive Systems Scenarios. *14th International Conference, CMC14*, pp.63-80, 2013.
19. S. Jie and X. Y. Liu. Density-Based Clustering by P System with Active Membranes on Commodity Recommendation in E-commerce Websites. *WSEAS Transactions on Computers*, v 13, p 20-33, 2014.
20. http://archive.ics.uci.edu/ml/.
21. J. S. Schlimmer. (1987). Concept Acquisition through Representational Adjustment (Technical Report 87-19). Doctoral dissertation, *Department of Information and Computer Science, University of California, Irvine.*