A K-nearest Based Clustering Algorithm by P Systems with Active Membranes

Jie Xue
School of Management Science and Engineering, Shandong Normal University, Jinan, China
Email:xiaozhuzhu1113@163.com

Xiyu Liu
School of Management Science and Engineering, Shandong Normal University, Jinan, China
Email:sdxyliu@163.com

***Abstract*—The purpose of this paper is to propose a new way to solving clustering problems, which combines membrane computing with a k-nearest based algorithm inspired by chameleon algorithm. The new algorithm is defined as PKNBA, which can obtain the k-nearest graphs, complete the partition of subgraph through communication rules, evolution rules, dissolution rules and division rules in P system with active membranes. The whole process of PKNBA algorithm is shown by a 10 points test data set, which indicates the feasibility and less time consuming of the algorithm. All the processes are conducted in membranes. Cluster results via the famous iris and wine data set verify that the proposed PKNBA algorithm can cluster data set more accurate than k-means algorithm. The influences of parameters to the algorithm are illustrated also. The PKNBA provides an alternative for traditional computing.**

***Index Terms*—k-nearest graph, subgraph partition, P systems, active membrane, membrane division, PKNBA algorithm**

I. Introduction

Membrane computing is a new branch of natural computing which is initiated by Păun at the end of 1998[1]. It abstracts computing models from the functioning of living cells, just like DNA computing coming from gene, particle swarm optimization from biomes, etc. Membrane computing deals with distributed and parallel computing models, processing multisets of symbol objects in a localized manner. Evolution rules and evolving objects are encapsulated into compartments defined by membranes. The communications between compartments and with the environment play an important role in the processes.

The advantage of these methods is the huge inherent parallelism. It has drawn great attention from the scientific community so far. The obtained computing systems proved to be so powerful that it is equivalent with Turing machines[2] even when using restricted combinations of features, and also computationally efficient. From now on, a number of applications were reported in several areas: biology, bio-medicine, linguistics, computer graphics, economics, approximate optimization, cryptography, etc [3].

The various types of membrane systems are known as P systems after Gheorghe Păun who first conceived the model. Membrane division is inspired from cell division well known in biology. P system with active membrane has the ability to do cell division by rules, which is a method to reduce time into line. So far, P system with active membrane is used for solving basis hard problems, typically NP-complete problems in polynomial (often, linear) time. Details can be found in [4,5,6]. Recently, PSPACE-complete problems were also attacked in this way (see [7]).

Clustering plays an essential and indispensable role in data mining. Although several methods are available in these areas [8], these algorithms exhibit polynomial or exponential complexity when the number of clusters is unknown and the data set is huge. It makes problems more challenging. Chameleon algorithm is a hierarchal clustering algorithm, which can find any shaped and high quality clusters compared with some famous clustering algorithm as BIRCH and DBSCAN [9].

Although membrane computing and cluster analysis receive much attention and rapid developed, there are rare combination of these two important research areas. Monica Cardona [10] presented an idea of using membrane computing to do the hierarchical clustering, where data are set into 0, 1 form, n + 1 membrane are used. Xiyu Liu [11] proposed a kind of P system on simplicial complexes to solve a grid based clustering problem.

Inspired by the researches above, this paper focuses on the joint study of membrane computing with cluster analysis. We use the k-nearest based clustering algorithm which is motivated by Chameleon algorithm. Then, we use membrane to conduct the whole process of computation, which can obtain the k-nearest graphs by evolution rules and communication rules, complete the partition of subgraph, output the final clustering result through communication rules, dissolution rules and division rules. A 10 points example is used to indicate the feasibility of the provided algorithm.Iris and wine benchmark data set demonstrates the algorithm's accuracy. The influence of parameters k and t is also

Corresponding author: Xiyu Liu,sdxyliu@163.com

doi:10.4304/jsw.9.3.716-725

illustrated.

II P SYSTEMS WITH ACTIVE MEMBRANES

A P system with active membranes (of degree $m \geq 1$) is a construct of the form[12][13]:

$$\prod = (\mathrm{O}, E, \mathrm{H}, \mu, \omega_1, \ldots, \omega_m, \mathrm{e}_1, \ldots, e_m, R, \mathrm{i}_0)$$

where O is the alphabet of objects. E = {0, . . . , n-1} with $n \geq 1$ is the set of electrical charges (polarizations). μ is a membrane structure with m membranes, labelled with elements of H, where H is a finite set of labels for membranes. $\omega_1,\ldots,\omega_m$ are strings over O indicating the multisets of objects at the beginning present in the m regions of μ . $\mathrm{e}_1,\ldots,e_m$ are the polarizations at the beginning assigned to the membranes 1, . . . , m. R is a finite set of rules of the following forms:

(a) evolution rules: $[a \to v]_h^i$
for $h \in H$, $i \in E$, $a \in O$, $v \in O^*$

(b) communication rules : $a[]_h^i \to [b]_h^j$
for $h \in H$, $i, j \in E$, $a,b \in O$

(c) communication rules: $[a]_h^i \to []_h^j b$
for $h \in H$, $i, j \in E$, $a,b \in O$

(d) membrane dissolution rules: $[a]_h^i \to b$
for $h \in H$, $i, j \in E$, $a,b \in O$

(e) division rules: $a[]_h^i \to [b]_h^j [c]_h^k$
for $h \in H$, $i, j,k \in E$, $a,b,c \in O$

Rule (a) is rewriting rule used in parallel in the region of membrane h, provided that the polarization of the membrane is i. (b) and (c) are communication rules, where (b) sends an object into a membrane, possibly changes the polarization of the membrane from i to j. On the contrary, (c) sends an object out of a membrane, possibly changes the polarization of the membrane also. (d) is dissolution rule, which is a special rule in P system with active membrane, allowing membrane's dissolution in reaction with an object. (e) is used for membrane division, in reaction with an object, the membrane is divided into two membranes with the same label, possibly of different polarizations. The object specified in the rule is replaced in the two new membranes by possibly new objects, and the remaining objects are duplicated.

A P system is called stable if, even if some rules are still applicable, their application does not change the string/object content of the membrane structure, nor the membrane structure itself.

It gives priority to rules in communication P system, for $\{R_{i1},\ldots R_{in}\}$ in membrane i in [14]. There is the priority that $R_{i1} \succ R_{i2} \succ,\ldots,\succ R_{in}$, which means if $R_{i1},\ldots R_{in}$ can be used all, R_{i1} will work firstly, then, $R_{i2}, R_{i3},\ldots, R_{i(n-1)}$, R_{in} will be used at last.

To expand the use of membrane computing, here we combine priority of rules with P system with active membranes.

III K-NEAREST BASED CLUSTERING ALGORITHM BY P SYSTEMS WITH ACTIVE MEMBRANES

A. *K-nearest Based Clustering Algorithm*

Chameleon algorithm is a hierarchal clustering algorithm that aims to find the k-nearest graph [8]. This algorithm has several advantages compared with other hierarchal algorithms, such as robust clustering (using links) and clustering (using representatives) algorithms. The Chameleon algorithm uses the relative interconnection RI (C_i, C_j) and the relative closeness RC (C_i,C_j) of each cluster (C_i, C_j) to determine their similarity

$$RI(C_i, C_j) = |EC_{C_i,C_j}| / (1/2 |EC_{C_i}| + |EC_{C_j}|)s \quad (1)$$

$$RC(C_i, C_j) = S_{EC_{C_i,C_j}} / (S_{EC_{C_i}} \times |C_i| / (|C_i| + |C_j|) + S_{EC_{C_j}} \times |C_j| / (|C_i| + |C_j|)) \quad (2)$$

Inspired by Chameleon algorithm[15], the k-nearest graph is used in this study to implement several components of the proposed model. The basic idea for spatial data clustering is to find the k-nearest neighbor graph for every data, cut edges with lengths beyond the threshold from the graph, and cluster patterns in the same subgraph.

To make data set being clustered suit for k-nearest based Algorithm. At first, we need to do some preparation for data. We define that data set is a k dimension set δ with n elements. Those n elements are the individuals who will be clustered. We use matrix to denote them[16]:

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \ldots & W_{1k} \\ W_{21} & W_{22} & W_{23} & \ldots & W_{2k} \\ \ldots & & & \ldots & \\ W_{n1} & W_{n2} & W_{n3} & \ldots & W_{nk} \end{bmatrix}$$

Where W_{ij} is the value of the j-th variable of the individual i.

Besides, similarity is very important in grouping data, thus establishing criteria to measure similarities of the data is necessary. Evidently, the clustering that can be obtained depends on the chosen similarity function. This function, called similarity, contains various measuring methods which are defined below.

Definition 2:

A distance between any two individual over the k dimension set $\delta = \{W_1, W_2, \ldots, W_n\}$ *is calculated by*

$$D_{ij}=\sqrt{(W_{i1}-W_{j1})^2+(W_{i2}-W_{j2})^2+.....+(W_{ik}-W_{jk})^2} \quad (3)$$

Hence the distance of the k dimension set δ is shown as a n*n matrix D, to adapt the calculation, we use D' to show distances between data, which is distances after take integer of D.

$$D'=\begin{bmatrix} 0 & D'_{12} & D'_{13} & \cdots & \cdots & D'_{1(n-2)} & D'_{1(n-1)} & D'_{1n} \\ & 0 & D'_{23} & \cdots & \cdots & D'_{2(n-2)} & D'_{2(n-1)} & D'_{2n} \\ & & & & \cdots & \cdots & \cdots & \cdots \\ & & & & & 0 & D'_{(n-2)(n-1)} & D'_{(n-2)n} \\ & & & & & & 0 & D'_{(n-1)n} \\ & & & & & & & 0 \end{bmatrix}$$

Definition 3:

A similarity S over the k dimension set $\delta=\{W_1,W_2,...,W_n\}$ *always can be defined as the distance between two individuals. Those elements which have a short distance are more similar than that with a long one. Therefore, here we use distance to denote similarity among individuals.*

$S(W_i,W_j)=D_{ij}$ *,S is symmetric as D, that is*:

$$S(W_i,W_j)=S(W_j,W_i), S(W_i,W_i)=S(W_j,W_j)=0$$

The k-nearest based algorithm use membrane structure as Fig.3.2, it put all similarities of each data i into membrane i ,through evolution rules and communication rules between membrane n+1 and membrane i , it find k-nearest graphs for every data i .In membrane i , rewriting rules divide the k-nearest graphs into subgraphs by threshold t given before calculation and obtain the clustering results. Finally, membrane i send the clustering result of every data i into output membrane n+1 through dissolution rules. The final result is read out from the membrane produced by division rules .The process of k-nearest based algorithm is as follows:

```
Begin
   Initialize the membrane structure;
Execute communication rules between n+1 and
membrane I in parallel
        w ← 1
 while ( w ≤ k ) do
    Execute evolution rules in membrane i  in parallel
      w ← w + 1
  End
Execute communication rules between n+1 and
membrane i in parallel
 for  z = 1, z ≤ t, z++
      Execute rewriting rules in membrane I in parallel
  End
   Execute communication rules between n+1 and
membrane i in parallel
   Execute dissolution rules in membrane1
   Execute division rule in n+1
   Output objects from membrane i0
End
```

B. P System Design for K-nearest Based Algorithm

The purpose of this paper is to obtain the cluster result of a k dimension set δ of n different individuals by P Systems with active membrane. P Systems with active membrane design for k-nearest based algorithm(PKNBA) as is Fig.3.1 a tuple:

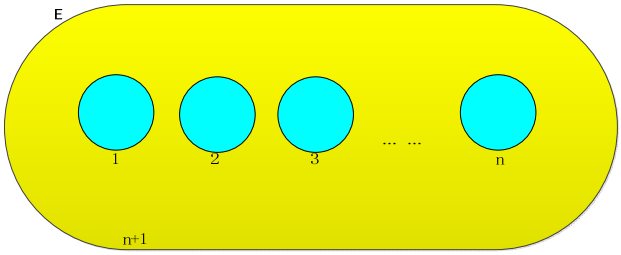

Figure 1. membrane structure of P Systems with active membrane design for K-nearest based algorithm

$$\prod=(O,E,H,\mu,\omega_1,\ldots,\omega_m,e_1,\ldots,e_m,R,i_0)$$

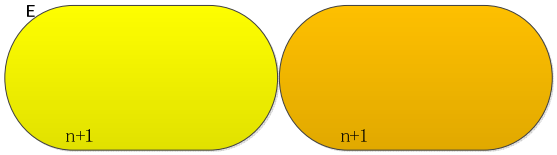

Figure 2. final status of membrane structure of PKNBA

,where

(1) $O=\{a_{ij},A_{ij},B_{ij},s,h,u,v,w,x,y,z\}$,
$1\le i\le n, 1\le j\le n, i\ne j$

(2) $E=\{0,1,2\}$

(3) $H=\{0,1,2,...,n\}$

(4) $\mu=[[]_1[]_2[]_3,...,[]_i,...,[]_n]_{n+1}$

(5) $\omega_i=\{a_{ij}^{d_{ij}} \mid 1\le i\le n, 1\le j\le n, i\ne j\}$
$\omega_{n+1}=\{u,y\}$

(6) $e_1=e_2...e_n=0$;
$e_{n+1}=0$

(7) R:

$u[]_i^0\to[v]_i^0$;

$[v^{d_{ij}}a_{ij}^{d_{ij}}\to v^{d_{ij}}A_{ij}^{d_{ij}}w]_i^0$;

$[w^k]_i^0\to x[]_i^1$;

$y[]_i^1\to[z]_i^1$;

$[z^{d_{ij}}A_{ij}^{d_{ij}}\to z^{d_{ij}}B_{ij}^{d_{ij}}]_i^1$;

$[z^t\to s]_i^1$;

$[z^{d_{ij}}A_{ij}^{d_{ij}}\to z^{d_{ij}}B_{ij}^{d_{ij}}]_i^1\succ[z^t\to s]_i^1$

($d_{ij} = t$);

$[s]_i^1 \to h[]_i^2$;

$[B_{ij}{}^{d_{ij}}]_i^2 \to B_{ij}{}^{d_{ij}}$;

$[B_{ij}{}^{d_{ij}} B_{ji}{}^{d_{ji}} \to B_{ij}{}^{d_{ij}}]_{n+1}^0$;

$[B_{1j}{}^{d_{1j}} B_{2j}{}^{d_{2j}} ... B_{nj}{}^{d_{nj}} uxyh]_{n+1}^0 \to$

$[B_{1j}{}^{d_{1j}} B_{2j}{}^{d_{2j}} ... B_{nj}{}^{d_{nj}}]_{n+1}^1 [uxyh]_{n+1}^2$

(8) $i_0 = []_{n+1}^1$ will output the clustering result as Fig.3.2

In the initial configuration, all polarizations of membranes are 0, rule $u[]_i^0 \to [v]_i^0$ is activated firstly since 'u' is initialized object in membrane n+1, 'v' is sent into membrane i continuously in parallel, which stands for every point. When the number of 'v' reaches the minimum number of distance d_{ij} in every membrane i, $a_{ij}{}^{d_{ij}}$ will change into $A_{ij}{}^{d_{ij}}$ and one 'w' is produced by the action of rule $[v^{d_{ij}} a_{ij}{}^{d_{ij}} \to v^{d_{ij}} A_{ij}{}^{d_{ij}} w]_i^0$ (every membrane does not activate their rule $[v^{d_{ij}} a_{ij}{}^{d_{ij}} \to v^{d_{ij}} A_{ij}{}^{d_{ij}} w]_i^0$ simultaneously) .These processes will go on working until there are k 'w' obtained and $[w^k]_i^0 \to x[]_i^1$ will be switched on, the polarization of membrane i changes into 1 from 0, $u[]_i^0 \to [v]_i^0$ can not work any more. Here we obtain k-nearest graph of each points in every membrane i.

Then, the mission is to cut k-nearest graph into subgraph and get the clustering result. $y[]_i^1 \to [z]_i^1$ will work in polarization 1 and one 'z' is put into membrane i, the strategy of our algorithm is to erase those distances which are beyond threshold t, it can be transformed into getting those distances within threshold t. Thus, $[z^{d_{ij}} A_{ij}{}^{d_{ij}} \to z^{d_{ij}} B_{ij}{}^{d_{ij}}]_i^1$ is used to receive satisfied distances $B_{ij}{}^{d_{ij}}$ until the number of 'z' arrive at t, $[z^t \to s]_i^1$ is activated and 'z' is transformed into 's', it is worth to say that when $d_{ij} = t$ is achieve, $[z^{d_{ij}} A_{ij}{}^{d_{ij}} \to z^{d_{ij}} B_{ij}{}^{d_{ij}}]_i^1 \succ [z^t \to s]_i^1$,which means $[z^{d_{ij}} A_{ij}{}^{d_{ij}} \to z^{d_{ij}} B_{ij}{}^{d_{ij}}]_i^1$ takes precedence over $[z^t \to s]_i^1$, then, $[s]_i^1 \to h[]_i^2$ changes polarization of membrane i from 1 to 2, 's' into 'h'. $y[]_i^1 \to [z]_i^1$ will lose its power .So far the clustering results is obtained.

The last step is to read out objects, all objects B_{ij} are sent out from membrane i by $[B_{ij}{}^{d_{ij}}]_i^2 \to B_{ij}{}^{d_{ij}}$ and membrane i is dissolved. Meanwhile, membrane division rule is used in membrane n+1, $[B_{ij}{}^{d_{ij}} uxyh]_{n+1}^0 \to [B_{ij}{}^{d_{ij}}]_{n+1}^1 [uxyh]_{n+1}^2$ makes all the objects we need together into the output membrane $[]_{n+1}^1$. Data not appearing in membrane i_0 are considered as outlines or can be put in a single cluster. The whole process of P-k-nearest algorithm is done.

C. An Overview of Computations

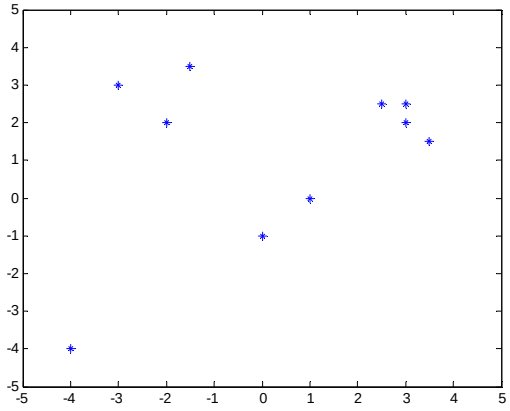

Figure 4. 10 points test data set

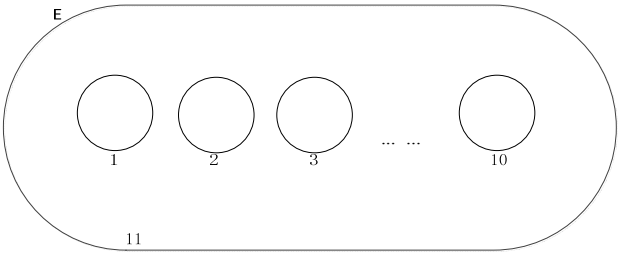

Figure 3. membrane structure for 10 points data set

In this subsection, we will show a simple example with ten data to certify the efficiency of our algorithm. Data set is shown in Fig.3.3 and the membrane structure is shown in Fig.3.4. K is 5, threshold t is 2.

The W^T of the matrix which stands for information of data is :

$$W^T = \begin{bmatrix} -4 & -3 & -2 & -1.5 & 0 & 1 & 2.5 & 3 & 3 & 3.5 \\ -4 & 3 & 2 & 3.5 & -1 & 0 & 2.5 & 2.5 & 2 & 1.5 \end{bmatrix}$$

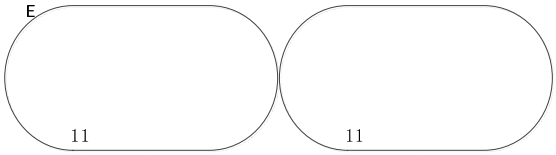

Figure 5. final status of membrane structure for 10 points data set

The distances between the ten individuals are as follows:

$$D' = \begin{bmatrix} 0 & 7 & 6 & 8 & 5 & 6 & 9 & 10 & 9 & 9 \\ 7 & 0 & 1 & 2 & 5 & 5 & 6 & 6 & 6 & 7 \\ 6 & 1 & 0 & 2 & 4 & 4 & 5 & 5 & 5 & 6 \\ 8 & 2 & 2 & 0 & 5 & 4 & 4 & 5 & 5 & 5 \\ 5 & 5 & 4 & 5 & 0 & 1 & 4 & 5 & 4 & 4 \\ 6 & 5 & 4 & 4 & 1 & 0 & 3 & 3 & 3 & 3 \\ 9 & 6 & 5 & 4 & 4 & 3 & 0 & 1 & 1 & 1 \\ 10 & 6 & 5 & 5 & 5 & 3 & 1 & 0 & 1 & 1 \\ 9 & 6 & 5 & 5 & 4 & 3 & 1 & 1 & 0 & 1 \\ 9 & 7 & 6 & 5 & 4 & 3 & 1 & 1 & 1 & 0 \end{bmatrix}$$

P Systems with active membrane design for K-nearest Based Algorithm for the ten data points is :

$\prod_{10} = (O, E, H, \mu, \omega_1, \ldots, \omega_{11}, e_1, \ldots, e_{11}, R, i_0)$

,where

(1) $O = \{a_{ij}, A_{ij}, B_{ij}, s, h, u, v, w, x, y, z\}$, where a_{ij}, A_{ij}, B_{ij} are shown in Table.

(2) $E = \{0,1,2\}$

(3) $H = \{0,1,2,\ldots,11\}$

(4) $\mu = [[]_1[]_2[]_3[]_4[]_5[]_6[]_7[]_8[]_9[]_{10}]_{11}$

(5) $\omega_1, \ldots, \omega_{10}$ are exhibited in Table.

$\omega_{11} = \{u, y\}$

(6) $e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7 = e_8 = e_9 = e_{10} = 0$;

$e_{11} = 0$

(7) $i_0 = []_{11}^1$

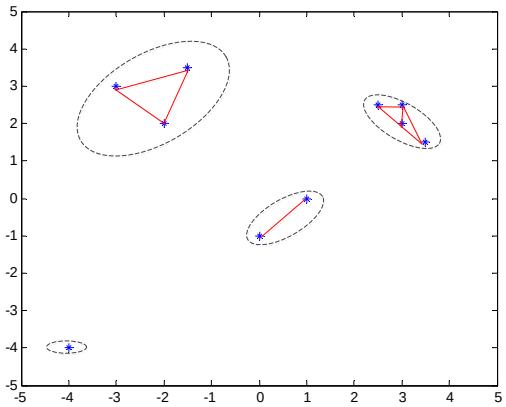

Figure 6. final clustering result of 10 points test data set

Rules execution processes are show in Table.3.1, according with rules actions, objects in membrane 1, 2,…10 ,11 and i_0 are changed as seen in Table.3.2. Fig.3.5 opened up the final status of membrane structure, Fig.3.6 obtains final results.

TABLE 3.1
THE PROCESS OF RULES EXECUTION OF PKNBA ALGORITHM OF 10 POINTS TEST DATA SET

Membrane	Rules
1	$u[]_1^0 \to [v]_1^0$, $u[]_1^0 \to [v]_1^0$, $u[]_1^0 \to [v]_1^1$, $u[]_1^0 \to [v]_1^0$, $u[]_1^0 \to [v]_1^0$, $[v^5 a_{15}{}^5 \to A_{15}{}^5 v^5 w]_1^0$, $u[]_1^0 \to [v]_1^0$, $[v^6 a_{13}{}^6 \to A_{13}{}^6 v^6 w]_1^0$, $[v^6 a_{16}{}^6 \to A_{16}{}^6 v^6 w]_1^0$, $u[]_1^0 \to [v]_1^0$, $[v^7 a_{12}{}^7 \to A_{12}{}^7 v^7 w]_1^0$, $u[]_1^0 \to [v]_1^0$, $[v^8 a_{14}{}^8 \to A_{14}{}^8 v^8 w]_1^0$, $[w^5]_1^0 \to x[]_1^1$, $y[]_1^1 \to [z]_1^1$, $[z^2 \to s]_1^1$, $[s]_1^1 \to h[]_1^2$
2	$u[]_2^0 \to [v]_2^0$, $[va_{23} \to A_{23} vw]_2^0$, $u[]_2^0 \to [v]_2^0$, $[v^2 a_{24}{}^2 \to A_{24}{}^2 v^2 w]_2^0$, $u[]_2^0 \to [v]_2^0$, $u[]_2^0 \to [v]_2^0$, $u[]_2^0 \to [v]_2^0$, $[v^5 a_{25}{}^5 \to A_{25}{}^5 v^5 w]_2^0$, $[v^5 a_{26}{}^5 \to A_{26}{}^5 v^5 w]_2^0$, $u[]_2^0 \to [v]_2^0$, $[v^6 a_{27}{}^6 \to A_{27}{}^6 v^6 w]_2^0$, $[w^5]_2^0 \to x[]_2^1$, $y[]_2^1 \to [z]_2^1$, $[zA_{23} \to zB_{23}]_2^1$, $y[]_2^1 \to [z]_2^1$, $[z^2 A_{24}{}^2 \to z^2 B_{24}{}^2]_2^1$, $[z^2 \to s]_2^1$, $[s]_2^1 \to h[]_2^2$, $[B_{23}]_2^2 \to B_{23}$, $[B_{24}{}^2]_2^2 \to B_{24}{}^2$
3	$u[]_3^0 \to [v]_3^0$, $[va_{32} \to A_{32} vw]_3^0$, $u[]_3^0 \to [v]_3^0$, $[v^2 a_{34}{}^2 \to A_{34}{}^2 v^2 w]_3^0$, $u[]_3^0 \to [v]_3^0$, $u[]_3^0 \to [v]_3^0$, $[v^4 a_{35}{}^4 \to A_{35}{}^4 v^4 w]_3^0$, $[v^4 a_{36}{}^4 \to A_{36}{}^4 v^4 w]_3^0$, $u[]_3^0 \to [v]_3^0$, $[v^5 a_{37}{}^5 \to A_{37}{}^5 v^5 w]_3^0$, $[w^5]_3^0 \to x[]_3^1$, $y[]_3^1 \to [z]_3^1$, $[zA_{32} \to zB_{32}]_3^1$, $y[]_3^1 \to [z]_3^1$, $[z^2 A_{34}{}^2 \to z^2 B_{34}{}^2]_3^1$, $[z^2 \to s]_3^1$, $[s]_3^1 \to h[]_3^2$, $[B_{32}]_3^2 \to B_{32}$, $[B_{34}{}^2]_3^2 \to B_{34}{}^2$
4	$u[]_4^0 \to [v]_4^0$, $u[]_4^0 \to [v]_4^0$, $[v^2 a_{42}{}^2 \to A_{42}{}^2 v^2 w]_4^0$, $[v^2 a_{43}{}^2 \to A_{43}{}^2 v^2 w]_4^0$, $u[]_4^0 \to [v]_4^0$, $u[]_4^0 \to [v]_4^0$, $[v^4 a_{46}{}^4 \to A_{46}{}^4 v^4 w]_4^0$, $[v^4 a_{47}{}^4 \to A_{47}{}^4 v^4 w]_4^0$, $u[]_4^0 \to [v]_4^0$, $[v^5 a_{45}{}^5 \to A_{45}{}^5 v^5 w]_4^0$, $[w^5]_4^0 \to x[]_4^1$, $y[]_4^1 \to [z]_4^1$, $y[]_4^1 \to [z]_4^1$, $[z^2 A_{42}{}^2 \to z^2 B_{42}{}^2]_4^1$, $[z^2 A_{43}{}^2 \to z^2 B_{43}{}^2]_4^1$, $[z^2 \to s]_4^1$, $[s]_4^1 \to h[]_4^2$, $[B_{42}{}^2]_4^2 \to B_{42}{}^2$, $[B_{43}{}^2]_4^2 \to B_{43}{}^2$

5	$u[]_5^0 \to [v]_5^0$, $[va_{56} \to A_{56}vw]_5^0$, $u[]_5^0 \to [v]_5^0$, $u[]_5^0 \to [v]_5^0$, $u[]_5^0 \to [v]_5^0$, $[v^4 {a_{53}}^4 \to {A_{53}}^4 v^4 w]_5^0$, $[v^4 {a_{57}}^4 \to {A_{57}}^4 v^4 w]_5^0$, $[v^4 {a_{59}}^4 \to {A_{59}}^4 v^4 w]_5^0$, $[v^4 {a_{510}}^4 \to {A_{510}}^4 v^4 w]_5^0$, $[w^5]_5^0 \to x[]_5^1$, $y[]_5^1 \to [z]_5^1$, $[zA_{56} \to zB_{56}]_5^1$, $[z^2 \to s]_5^1$, $[s]_5^1 \to h[]_5^2$, $[B_{56}]_5^2 \to B_{56}$
6	$u[]_6^0 \to [v]_6^0$, $[va_{65} \to A_{65}vw]_6^0$, $u[]_6^0 \to [v]_6^0$, $u[]_6^0 \to [v]_6^0$, $[v^3 {a_{67}}^3 \to {A_{67}}^3 v^3 w]_6^0$, $[v^3 {a_{68}}^3 \to {A_{68}}^3 v^3 w]_6^0$, $[v^3 {a_{69}}^3 \to {A_{69}}^3 v^3 w]_6^0$, $[v^3 {a_{610}}^3 \to {A_{610}}^3 v^3 w]_6^0$, $[w^5]_6^0 \to x[]_6^1$, $y[]_6^1 \to [z]_6^1$, $[zA_{65} \to zB_{65}]_6^1$, $[z^2 \to s]_6^1$, $[s]_6^1 \to h[]_6^2$, $[B_{65}]_6^2 \to B_{65}$
7	$u[]_7^0 \to [v]_7^0$, $[va_{78} \to A_{78}vw]_7^0$, $[va_{79} \to A_{79}vw]_7^0$, $[va_{710} \to A_{710}vw]_7^0$, $u[]_7^0 \to [v]_7^0$, $u[]_7^0 \to [v]_7^0$ $[v^3 {a_{76}}^3 \to {A_{76}}^3 v^3 w]_7^0$, $u[]_7^0 \to [v]_7^0$, $[v^4 {a_{74}}^4 \to {A_{74}}^4 v^4 w]_7^0$, $[w^5]_7^0 \to x[]_7^1$, $y[]_7^1 \to [z]_7^1$, $[zA_{78} \to zB_{78}]_7^1$, $[zA_{79} \to zB_{79}]_7^1$, $[zA_{710} \to zB_{710}]_7^1$, $[z^2 \to s]_7^1$, $[s]_7^1 \to h[]_7^2$, $[B_{78}]_7^2 \to B_{78}$, $[B_{79}]_7^2 \to B_{79}$, $[B_{710}]_7^2 \to B_{710}$
8	$u[]_8^0 \to [v]_8^0$, $[va_{87} \to A_{87}vw]_8^0$, $[va_{89} \to A_{89}vw]_8^0$, $[va_{810} \to A_{810}vw]_8^0$, $u[]_8^0 \to [v]_8^0$, $u[]_8^0 \to [v]_8^0$, $[v^3 {a_{86}}^3 \to {A_{86}}^3 v^3 w]_8^0$, $u[]_8^0 \to [v]_8^0$, $u[]_8^0 \to [v]_8^0$, $[v^5 {a_{83}}^5 \to {A_{83}}^5 v^5 w]_8^0$, $[w^5]_8^0 \to x[]_8^1$, $y[]_8^1 \to [z]_8^1$, $[zA_{87} \to zB_{87}]_8^1$, $[zA_{89} \to zB_{89}]_8^1$, $[zA_{810} \to zB_{810}]_8^1$, $[z^2 \to s]_8^1$, $[s]_8^1 \to h[]_8^2$, $[B_{87}]_8^2 \to B_{87}$, $[B_{89}]_8^2 \to B_{89}$, $[B_{810}]_8^2 \to B_{810}$
9	$u[]_9^0 \to [v]_9^0$, $[va_{97} \to A_{97}vw]_9^0$, $[va_{98} \to A_{98}vw]_9^0$, $[va_{910} \to A_{910}vw]_9^0$, $u[]_9^0 \to [v]_9^0$, $u[]_9^0 \to [v]_9^0$, $[v^3 {a_{96}}^3 \to {A_{96}}^3 v^3 w]_9^0$, $u[]_9^0 \to [v]_9^0$, $[v^4 {a_{95}}^4 \to {A_{95}}^4 v^4 w]_9^0$, $[w^5]_9^0 \to x[]_9^1$, $y[]_9^1 \to [z]_9^1$, $[zA_{97} \to zB_{97}]_9^1$, $[zA_{98} \to zB_{98}]_9^1$, $[zA_{910} \to zB_{910}]_9^1$, $[z^2 \to s]_9^1$, $[s]_9^1 \to h[]_9^2$, $[B_{97}]_9^2 \to B_{97}$, $[B_{98}]_9^2 \to B_{98}$, $[B_{910}]_9^2 \to B_{910}$
10	$u[]_{10}^0 \to [v]_{10}^0$, $[va_{107} \to A_{107}vw]_{10}^0$, $[va_{108} \to A_{108}vw]_{10}^0$, $[va_{109} \to A_{109}vw]_{10}^0$, $u[]_{10}^0 \to [v]_{10}^0$, $u[]_{10}^0 \to [v]_{10}^0$, $[v^3 {a_{106}}^3 \to {A_{106}}^3 v^3 w]_{10}^0$, $u[]_{10}^0 \to [v]_{10}^0$, $[v^4 {a_{105}}^4 \to {A_{105}}^4 v^4 w]_{10}^0$, $[w^5]_{10}^0 \to x[]_{10}^1$, $y[]_{10}^1 \to [z]_{10}^1$, $[zA_{107} \to zB_{107}]_{10}^1$, $[zA_{108} \to zB_{108}]_{10}^1$, $[zA_{109} \to zB_{109}]_{10}^1$, $[z^2 \to s]_{10}^1$, $[s]_{10}^1 \to h[]_{10}^2$, $[B_{107}]_{10}^2 \to B_{107}$, $[B_{108}]_{10}^2 \to B_{108}$, $[B_{109}]_{10}^2 \to B_{109}$
11	$[B_{23}B_{32} \to B_{23}]_{11}^0$, $[{B_{24}}^2 {B_{42}}^2 \to {B_{24}}^2]_{11}^0$, $[{B_{34}}^2 {B_{43}}^2 \to {B_{43}}^2]_{11}^0$, $[B_{56}B_{65} \to B_{56}]_{11}^0$, $[B_{78}B_{87} \to B_{78}]_{11}^0$, $[B_{79}B_{97} \to B_{79}]_{11}^0$, $[B_{710}B_{107} \to B_{710}]_{11}^0$, $[B_{89}B_{98} \to B_{89}]_{11}^0$, $[B_{810}B_{108} \to B_{810}]_{11}^0$, $[B_{910}B_{109} \to B_{910}]_{11}^0$, $[B_{23}{B_{24}}^2 {B_{34}}^2 B_{56}B_{78}B_{79}B_{710}B_{89}B_{810}B_{910}uxyh]_{11}^0 \to [B_{23}{B_{24}}^2 {B_{34}}^2 B_{56}B_{78}B_{79}B_{710}B_{89}B_{810}B_{910}]_{11}^1 [uxyh]_{11}^2$

TABLE 3.2

OBJECTS CHANGES OF PKNBA ALGORITHM OF 10 POINTS TEST DATA SET

membrane	Initial status	K-nearest graph	Final status
1	${a_{12}}^7, {a_{13}}^6, {a_{14}}^8, {a_{15}}^5, {a_{16}}^6, {a_{17}}^9, {a_{18}}^{10}, {a_{19}}^9, {a_{110}}^9$	${A_{15}}^5, {A_{13}}^6, {A_{16}}^6, {A_{12}}^7, {A_{14}}^8$	
2	${a_{21}}^7, a_{23}, {a_{24}}^2, {a_{25}}^5, {a_{26}}^5, {a_{27}}^6, {a_{28}}^6, {a_{29}}^6, {a_{210}}^7$	$A_{23}, {A_{24}}^2, {A_{25}}^5, {A_{26}}^5, {A_{27}}^6$	
3	${a_{31}}^6, a_{32}, {a_{34}}^2, {a_{35}}^4, {a_{36}}^4, {a_{37}}^5, {a_{38}}^5, {a_{39}}^5, {a_{310}}^6$	$A_{32}, {A_{34}}^2, {A_{35}}^4, {A_{36}}^4, {A_{37}}^5$	
4	${a_{41}}^8, {a_{42}}^2, {a_{43}}^2, {a_{45}}^5, {a_{46}}^4, {a_{47}}^4, {a_{48}}^5, {a_{49}}^5, {a_{410}}^5$	${A_{42}}^2, {A_{43}}^2, {A_{45}}^5, {A_{46}}^4, {A_{47}}^4$	

5	$a_{51}^{5}, a_{52}^{5}, a_{53}^{4}, a_{54}^{5}, a_{56}, a_{57}^{4}, a_{58}^{5}, a_{59}^{4}, a_{510}^{4}$	$A_{53}^{4}, A_{56}, A_{57}^{4}, A_{59}^{4}, A_{510}^{4}$	
6	$a_{61}^{6}, a_{62}^{5}, a_{63}^{4}, a_{64}^{4}, a_{65}, a_{67}^{3}, a_{68}^{3}, a_{69}^{3}, a_{610}^{3}$	$A_{65}, A_{67}^{3}, A_{68}^{3}, A_{69}^{3}, A_{610}^{3}$	
7	$a_{71}^{9}, a_{72}^{6}, a_{73}^{5}, a_{74}^{4}, a_{75}^{4}, a_{76}^{3}, a_{78}, a_{79}, a_{710}$	$A_{74}^{4}, A_{76}^{3}, A_{78}, A_{79}, A_{710}$	
8	$a_{81}^{10}, a_{82}^{6}, a_{83}^{5}, a_{84}^{5}, a_{85}^{5}, a_{86}^{3}, a_{87}, a_{89}, a_{810}$	$A_{83}^{5}, A_{86}^{3}, A_{87}, A_{89}, A_{810}$	
9	$a_{91}^{9}, a_{92}^{6}, a_{93}^{5}, a_{94}^{5}, a_{95}^{4}, a_{96}^{3}, a_{97}, a_{98}, a_{910}$	$A_{95}^{4}, A_{96}^{3}, A_{97}, A_{98}, A_{910}$	
10	$a_{101}^{9}, a_{102}^{7}, a_{103}^{6}, a_{104}^{5}, a_{105}^{4}, a_{106}^{3}, a_{107}, a_{108}, a_{109}$	$A_{105}^{4}, A_{106}^{3}, A_{107}, A_{108}, A_{109}$	
11	y,u		$B_{23}, B_{56}, B_{78}, B_{79}, B_{710}, B_{89}, B_{810}, B_{910}, B_{24}^{2}, B_{34}^{2}$

IV Experiments and Discussion

In this section, we present some examples to illustrate the performance of our algorithm. Then we will show that our technique gives clusters more naturally.

A *Example One*

k-nearest based algorithm by P system with active membrane proposed in this study was applied to two UCI data set Iris and Wine. Information of these two data set are as Table below.

The iris data set is one of the most popular data set. However, it is famous for its difficulty on clustering because the data in this set are alternative and do not have any obvious bounds. It has three classification: iris-virginica, iris-versicolor and iris-setosa

The wine data set was derived from three different grape wines in Italy, including 178 records.

Each grape wine comprises 13 dimensions of wine properties.

The data set was divided into three groups with the number of setosa, versicolour, and virginica are59,48,71.

Wine and iris data were used to verify the feasibility and validity of the proposed algorithm [17]. The correct rate of iris and wine clustering by k-means is 82.7% and 64.0% in Table.4.2 and 4.3. It is obvious that the proposed algorithm is more useful and correct.

TABLE1

Information of Iris and Wine data set

Data set	number	dimension	group
Iris	150	4	3
Wine	178	13	3

TABLE 2.

CLUSTERING CORRECT RATE OF IRIS

algorithm	Cluster 1	Cluster 2	Cluster 3	Correct rate
PKNBA	15:50	6:50	0:50	86.0%
k-means	12:50	8:50	6:50	82.7%

This data set should be divided into three clusters.

TABLE 3.

CLUSTERING CORRECT RATE OF WINE

algorithm	Cluster 1	Cluster 2	Cluster 3	Correct rate
PKNBA	0:59	11:71	10:48	88.8%
k-means	22:59	36:71	4:48	64.0%

B *Example two*

Next, we will show an simple example to certify the influence of parameters k and t in our algorithm. Data set is shown in Fig.4.1

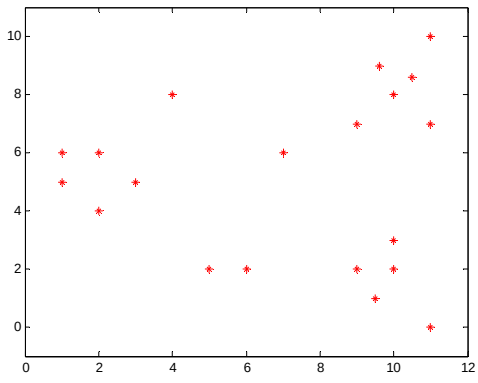

Figure 7. a 20 points data set

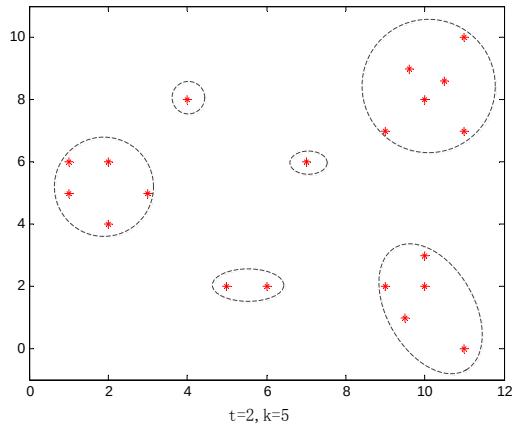

Figure 8. clustering results with t=2

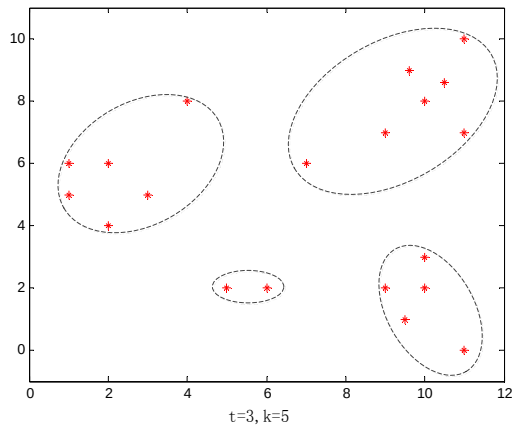

Figure9. clustering results with t=3

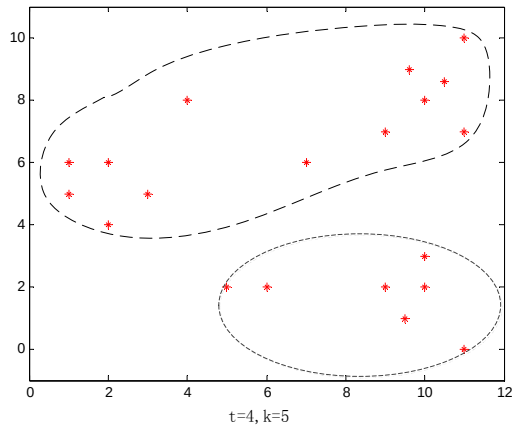

Figure 10. clustering results with t=4

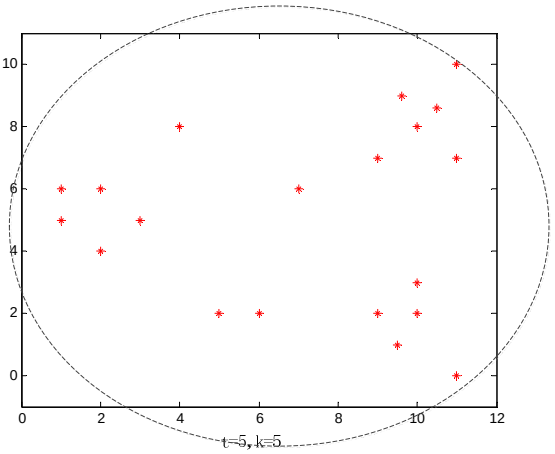

Figure 11. clustering results with t=5

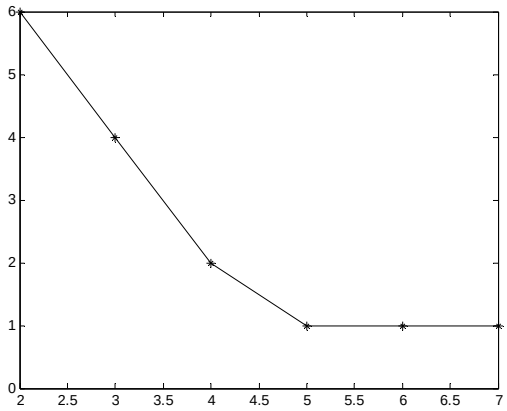

Figure 12. cluster change by the influence of t

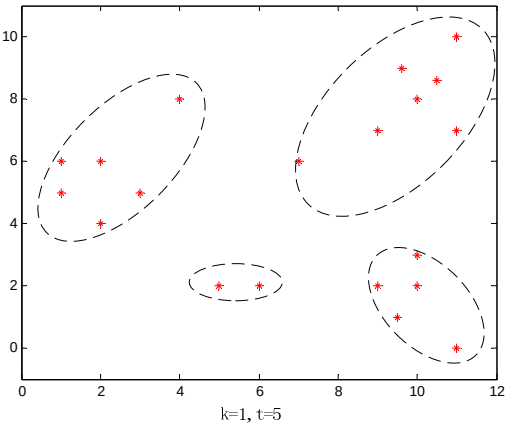

Figure 13. clustering results with k=1

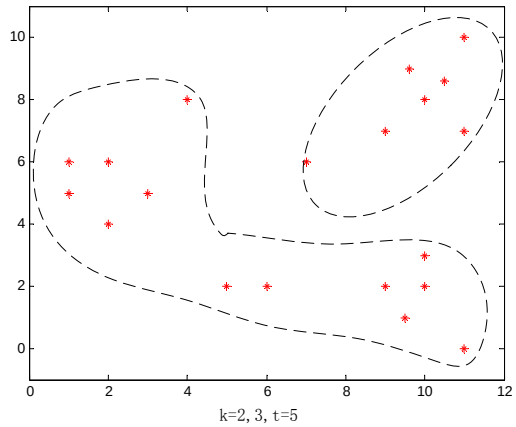

Figure 14. clustering results with k=2 and 3

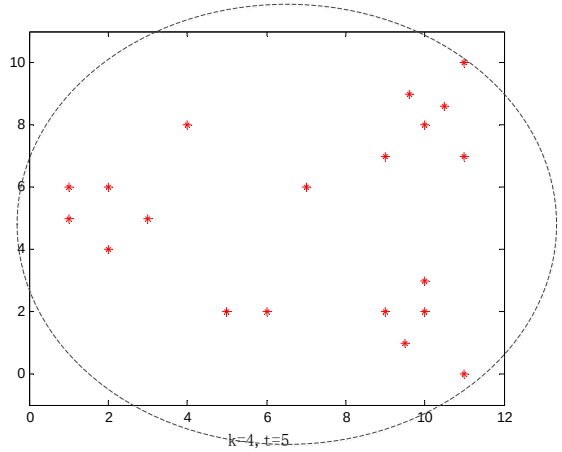

Figure 15. clustering results with k=4

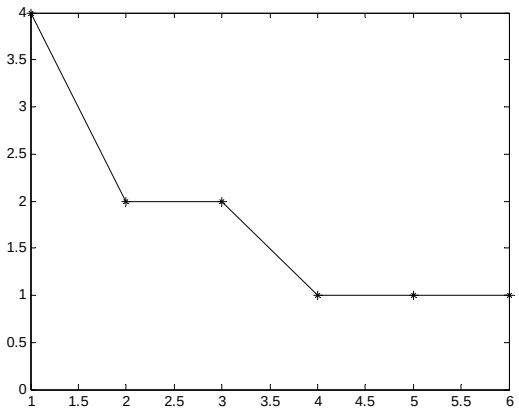

Figure 16. cluster change by the influence of k

Fig.4.2~4.5 and Fig.4.6 indicate the influence of threshold t to the clustering results. At the beginning, t increasing leads to the reduction of group numbers. when t arrives at 4, group number maintains in 1 and does not change any more.

Fig.4.7~4.9 and Fig.4.10 demonstrate the influence of k to the clustering results. Firstly, k increasing leads to the reduction of group numbers as well as t. when k arrives at 4, group number maintains in 1 and does not change any longer.

We can see that the effect of t is greater than k.

P-Lingua is a programming language for membrane computing which aims to be a standard to define P systems. One of its main features is to remain as close as possible to the formal notation used in the literature to define P systems. P-Lingua is also the name of a software package that includes several built-in simulators for each supported model as well as the needed compilers to simulate P-Lingua programs[18]. Our experiments can be implemented by P-Lingua 2.0 in Win7, 32bits, core i3,eclipse 4.2.

C Analysis of PKNBA Algorithm

TABLE 4

TIME COMPLEXITY OF P-CHAMELEON ALGORITHM

data	specific status	Time complexity
best	k	O(n)
worst	n+kn	O(n)
common	g+θn	O(n)

It is shown by table above, the best time complexity of PKNBA algorithm is O(n) and the worst time complexity is also O(n). Finding k-nearest graph consume g($k \le g \le n$), subgraph partition uses θn steps , where θ is approximate to every data's division steps.

Hence, we can conclude that the time complexity of the k-nearest based algorithm by P system with active membrane is O(n), which reduce the time complexity in some degree.

V CONCLUSION

A new strategy for the clustering algorithm using membrane computing is proposed in this paper. The P system with active membrane is used to implement clustering. All the processes are conducted in membranes. The whole process of the proposed k-nearest based clustering algorithm is shown by a 10 points test data set. Cluster result via the famous Iris and Wine data set verifies that the proposed algorithm can cluster data set more accurate than k-means algorithm. 20 points data set also illustrates the changes obtained from the chosen values of *k* and threshold *t* assignment.

Although the process of the proposed algorithm is provided and a number of instances to prove

its feasibility are presented, there are also many works needed to do for further study. The proposed algorithm can also solve spatial data. However, it does not illustrate this aspect clearly. In the future, we will continue researching the use of membrane computing techniques to cluster three-dimensional or spatial data.

ACKNOWLEDGMENT

This research is supported by the Natural Science Foundation of China（no.61170038), the Natural Science Foundation of Shandong Province(no. ZR2011FM001）. Humanities and Social Sciences project Supported by Chinese Ministry of Education(12YJA630152)

REFERENCES

[1] Gheorghe Păun, A quick introduction to membrane computing, *The Journal of Logic and Algebraic Programming*, 79, 2010, pp. 291-294.

[2] Păun G, Rozenberg G, and Salomaa A, Membrane Computing, *Oxford University Pres*s, New York, 2010.

[3] Jinhui Zhao, Ning Wang, A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling, *Computers and Chemical Engineering*, 35 ,2011,pp.272-283.

[4] Gh. Păun, P systems with active membranes: attacking NP-complete problems, J. Automata, Lang. *Combin*. 6 ,1, 2001, pp.75–90.

[5] G.h. Păun, Membrane Computing: An Introduction, Springer, Heidelberg, 2002,

[6] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, Teoría de la Complejidad en Modelos de Computation Celular con Membranas, *Editorial Kronos*, Sevilla, 2002.

[7] P. Sosik, Solving a PSPACE-complete problem by P systems with active membranes, In: M. Cavaliere, C. Marín-Vide, Gh.Păun (Eds.), Proceedings of the Brainstorming Week on Membrane Computing. *Report GRLMC 26/03*, 2003, pp. 305–312

[8] Han J. and M. Kamber, Data Mining, Concepts and Techniques, *Higher Education Press,* Morgan Kaufmann Publishers, Beijing, 2002.

[9] M.Acoub, F. Badran, S. Thiria. A topological hierarchical clustering: Application to ocean color classification, Lecture Notes in Computer Science, 2130 (2001),pp. 492-499.

[10] Mónica Cardona, Angels Colomer, Mario J. Peréz-Jiménz, Alba Zaragoza, Hierarchical Clustering with Membrane, *Computing*, pp.185-204.

[11] Xiyu Liu, Alice Xue, Communication P Systems on Simplicial Complexes with Applications in Cluster Analysis, *Discrete Dynamics in Nature and Society* , 2012.pp.1-17.

[12] Linqiang Pan, C. Marín-Vide Further remark on P systems with active membranes and two polarizations, *J. Parallel Distrib. Comput.* 66 ,2006,pp.867 – 872.

[13] Linqiang Pan, C. Marín-Vide Solving multidimensional 0-1 knapsack problem by P systems with input and active membranes, J. *Parallel Distrib. Comput.* 66 ,2006,pp.867 – 872.

[14] Rodica CETERCHI, Carlos MARTíN-VIDE,P Systems with Communication for Static Sorting, *GRLMC Report (M. Cavaliere, C. Martín-Vide, Gh. Paun, eds.), Rovirai Virgili Universit*y, 2003, pp.101-117

[15] Peng Bi, Study on application of improved Chameleon hierarchical Clustering algorithm in target clustering, *ZhejiangUniversity,*2009.

[16] Xue Jie, Liu Xiyu. Spatial Cluster by Communication P System on Trees, *Journal of information and computational science*, 9,13,2012, pp.3883-3893.

[17] Wang Yun, Research om clustering algorithm based on quantum theory, *Zhejiang university of technology*,2011.

[18] L. Diez Dolinski, R. Núñez Hervás, M. Cruz Echeandía, A. Ortega. Distributed Simulation of P Systems by Means of Map-Reduce：First Steps with Hadoop and P-Lingua. *IWANN 2011*, Part I, LNCS 6691, 2011. pp. 457–464.

[19] Juanying Xie, Shuai Jiang, Weixin Xie, Xinbo Gao,An Efficient Global K-means Clustering Algorithm, Journal of Computers, Vol 6, No 2 (2011), 271-279.

[20] Yu Zong, Ping Jin, Dongguan Xu, Rong Pan, A Clustering Algorithm based on Local Accumulative Knowledge Journal of Computers, Vol 8, No 2 (2013), 365-371.

[21] Qiu-yu Zhang, Peng Wang, Hui-juan Yang, Applications of Text Clustering Based on Semantic Body for Chinese Spam Filtering, Journal of Computers, Vol 7, No 11 (2012), 2612-2616.

Jie Xue is currently a PH.D. in management science and engineering of Shandong Normal University, China. Her

research interests include Data mining, Artificial Intelligence, biocomputing, distributed systems

Xiyu Liu received his Ph.D degree in math from Shandong University , China .He is currently a professor of management science and engineering at Shandong Normal University, China. His current research interests include Data mining, Artificial Intelligence, bio-computing, distributed systems