ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **17**, Number 1, 2014, 76–88

DE-MC: A Membrane Clustering Algorithm Based on Differential Evolution Mechanism

Hong PENG[1], Jiarong ZHANG[1], Yang JIANG[1], Xiaoli HUANG[1], Jun WANG[2]

[1] Center for Radio Administration and Technology Development, Xihua University, Chengdu, 610039, China

E-mail: ph.xhu@hotmail.com

[2] School of Electrical and Information Engineering, Xihua University, Chengdu, Sichuan, 610039, China

Abstract. A clustering algorithm using the framework of membrane computing is proposed in this paper. The P system used is a cell-like P system of two-layer nested structure: a skin membrane contains several elementary membranes. Each object in elementary membranes represents a group of cluster centers. Objects in the system evolve by using the differential evolution mechanism, and then the global optimal object in the skin membrane is updated by the best objects in all elementary membranes. The cell-like P system can automatically find the best cluster centers for a data set. The proposed DE-MC algorithm is evaluated on an artificial data set and a real-life data set and is further compared with classical k-means algorithm, GA-based clustering algorithm and DE-based clustering algorithm, respectively. The comparison results reveal that the proposed DE-MC algorithm is superior to the other three clustering algorithms in terms of clustering quality and robustness.

Key-words: Membrane computing; P systems; Clustering algorithm; Differential evolution.

1. Introduction

Clustering can be defined as the optimal partitioning of a given set of m data points into c classes such that data points belonging to the same class are as similar to each other as possible, whereas data points from two different classes share the maximum difference [1]. Clustering algorithms have been widely used in various fields, such as data mining, pattern recognition, image processing, and so on. Clustering algorithms can be hierarchical or partitional [2]. In hierarchical clustering, the output is a tree showing a sequence of clusterings with each cluster being a partition of the data set. Hierarchical clustering algorithms can be agglomerative (bottom-up) or divisive (top-down). Partitional clustering attempts to decompose the data set directly into a set of disjoint clusters. They try to optimize certain criteria (for example, a square error function). The criterion function may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measures of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters. An extensive survey of various clustering techniques can be found in [3]. A widely used partitional clustering algorithm is k-means, which attempts to directly decompose a data set into several disjoint clusters based on some criteria. However, k-means algorithm is sensitive to the initial clustering centers and easily falls into local minimal solutions. Moreover, when the number of data points is large, it takes an enormous time to find the global optimal solution. In order to overcome this problem, a lot of studies have been done in clustering. Krishna et al. have presented a method called genetic k-means algorithm for clustering analysis [4]. Mualik et al. have proposed a genetic algorithm based method to solve the clustering problem and experimented it on synthetic and real life data sets to evaluate its performance [5]. Laszlo et al. have presented a genetic algorithm that exchanges neighboring centers for k-means clustering [6]. Shelokar et al. have introduced an evolutionary algorithm based on ACO (ant colony optimization) algorithm for clustering problem [7]. Niknam et al. have presented a hybrid evolutionary optimization algorithm based on the combination of ACO and SA (simulated annealing) to solve the clustering problem [8]. An efficient hybrid approach based on PSO (particle swarm optimization), ACO and k-means for cluster analysis has been discussed in [9].

Membrane computing, initiated by Gh. Păun [10], is inspired from the structure and functioning of living cells as well as interaction of living cells in tissues and organs. Membrane computing investigates distributed parallel computing models, also known as P systems. Generally, they are characterized by three elements: (i) membrane structure, (ii) multisets and (iii) evolution rules. The multisets of objects are placed in compartments surrounded by membranes, and evolved by some given rules [11]. Roughly, P systems can be categorized into three classes: cell-like P systems, tissue-like P systems and neural-like P systems. Cell-like P systems have been studied first, and their membranes are arranged as a rooted tree. Tissue-like P systems are inspired from intercellular communication and cooperation between cells in tissues. Neural-like P systems are another type of P systems. Spiking neural P systems, as a class of neural-like P systems, have been widely studied in recent years. Over these years, a

large number of variants have been considered [12, 13, 14, 15, 16, 17, 18, 19, 20]. A variety of inherent mechanisms and characteristics that P systems possess are able to provide a new incentive for the research of clustering analysis.

In this paper, in order to overcome the k-means shortcomings, a novel membrane clustering algorithm integrating the differential evolution mechanism, called DE-MC algorithm, is proposed. The DE-MC algorithm is based on a cell-like P system of two-layer nested structure. Each object in membranes represents a group of cluster centers. The differential evolution mechanism of the known differential evolution algorithm is introduced as evolution rules in membranes. The main contribution of this paper is the presentation of a new partitional clustering algorithm within the framework of membrane computing to solve the clustering problem.

The rest of paper is organized as follows. The proposed DE-MC algorithm is discussed in detail in Section 2. Section 3 presents experimental results to illustrate the efficiency of the algorithm. Finally, Section 4 includes the conclusions.

2. The proposed DE-MC algorithm

In this work, we develop a novel partitional clustering algorithm using the framework of membrane computing, called the DE-MC algorithm, which integrates the differential evolution mechanism in the known differential evolution algorithms [21, 22]. Usually, a cell-like P system contains several membranes with a nested structure, and each membrane has some objects and evolution rules. The objects in membranes are transformed by using the evolution rules. The proposed DE-MC algorithm is based on a cell-like P system of two-layer nested structure. The cell-like P system and its main components are described in detail as follows.

2.1. Membrane structure

Our P system has a two-layer membrane structure, which consists of a skin membrane and q elementary membranes, shown in Fig. 1. The q elementary membranes are labeled by $1, 2, \ldots, q$ respectively and the skin membrane is labeled by 0. In the designed cell-like P system, each elementary membrane contains n objects, while the skin membrane has only one object that expresses the global best object found so far in the whole system during the computation. Evolution rules are contained in the q elementary membranes, so the elementary membranes are called evolution membranes. The skin membrane stores the global best object without evolution rules.

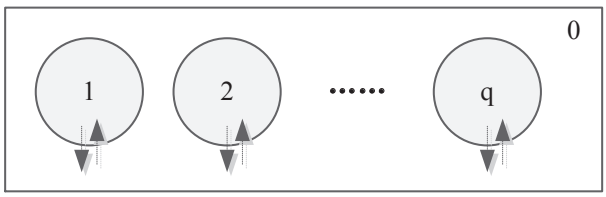

Fig. 1. The membrane structure of the designed cell-like P system.

2.2. Objects

Each object of the elementary membranes represents a group of candidate cluster centers. The cell-like P system searches for the optimal cluster centers for a data set by evolving objects in the elementary membranes. Assume that $D = \{x_i \in R^d \mid i = 1, 2, \ldots, m\}$ is a data set to be clustered, which is divided into k clusters, $C_1, C_2, \ldots, C_k$. The cluster centers of the k clusters are denoted by $z_1, z_2, \ldots, z_k$ respectively, where $z_i \in R^d$ $(i = 1, 2, \ldots, k)$. According to the principle of partitional clustering, if the Euclidean distances $d(x_i, C_p)$ $(p = 1, 2, \ldots, k)$ of a sample point x_i to k clusters $C_1, C_2, \ldots, C_k$ satisfy

$$d(x_i, C_j) = \min_{p=1,2,\ldots,k} \{d(x_i, C_p)\}, \tag{1}$$

then the sample point x_i is assigned to the center C_j. Thus, each object in the elementary membranes, which expresses a group of candidate cluster centers, should be a $(k \times d)$-dimensional vector of the form

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

where $(z_{i1}, z_{i2}, \ldots, z_{id})$ corresponds to the ith cluster center z_i. Figure 2 shows the object representation in the cell-like P system.

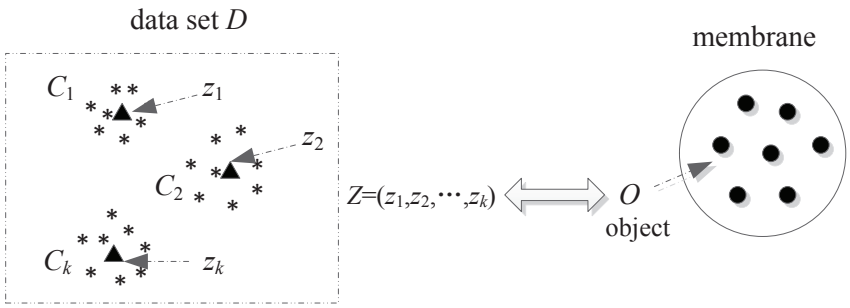

Fig. 2. The representation of objects in membranes.

2.3. Evolution rules

In order to obtain the optimal cluster centers, the cell-like P system uses evolution rules to evolve the objects in the elementary membranes. In this work, the differential evolution mechanism of differential evolution algorithm [21, 22] is introduced as the evolution rules of the elementary membranes, which consists of mutation, crossover and selection operators. Therefore, the cell-like P system has the evolution rules of three types: mutation rules, crossover rules and selection rules. For the jth object of ith elementary membrane, $Z_{i,j}$, the evolution rules are described as follows:

- Mutation rule ("DE/best/1/bin" type [22]). For the jth object of the ith elementary membrane, $Z_{i,j}$, the mutation rule evolves the object to generate a mutation object $U_{i,j}$:

$$U_{i,j} = Z_{best} + F(Z_{i,r_1} - Z_{i,r_2}) \tag{2}$$

  where Z_{i,r_1} and Z_{i,r_2} are two objects selected randomly from the ith elementary membrane, Z_{best} is the best global object in the skin membrane, and F is the scaling factor.

- Crossover rule. Based on the mutation object $U_{i,j}$ and the original object $Z_{i,j}$, a crossover object $V_{i,j}$ is generated by the crossover rule as follows:

$$V_{i,j,s} = \begin{cases} U_{i,j,s}, & \text{if rand}_s(0,1) \leq CR \text{ or } s = \text{rand}(j) \\ Z_{i,j,s}, & \text{otherwise} \end{cases} \tag{3}$$

  where $s = 1, 2, \ldots, k \times d$, $\text{rand}_s(0,1)$ is a random number from the interval [0,1], $CR \in [0,1]$ is a predefined crossover constant, and $\text{rand}(j) \in \{1, 2, \ldots, k \times d\}$ is a randomly selected subscript, which is used to ensure that $V_{i,j}$ gets at least one component from $Z_{i,j}$.

- Selection rule. Based on the fitness of the original object $Z_{i,j}$ and the crossover object $V_{i,j}$, the object that has better fitness value in both $Z_{i,j}$ and $V_{i,j}$ is selected as the object of the elementary membrane in the next computing step, $Z'_{i,j}$,

$$Z'_{i,j} = \begin{cases} V_{i,j}, & \text{if } f(V_{i,j}) \leq f(Z_{i,j}) \\ Z_{i,j}, & \text{otherwise} \end{cases} \tag{4}$$

  where $f(\cdot)$ is the fitness function to be minimized.

Figure 3 illustrates the evolution procedure of an object in elementary membranes during a computing step.

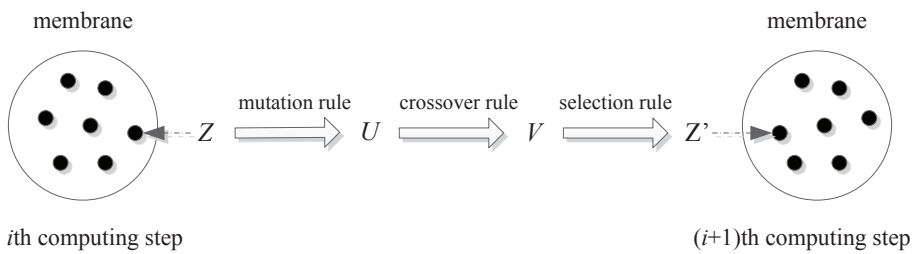

Fig. 3. The evolution procedure of an object in elementary membranes.

2.4. Communication rules

In addition to evolution rules, the cell-like P system uses communication rules to exchange and share objects between evolution membranes and skin membrane. The

skin membrane stores the global best object found so far in the whole system during the computation. After all the objects of the evolution membranes are evolved by differential evolution rules, the best object in each evolution membrane will update the global best object in the skin membrane. When the system halts, the global best object in the skin membrane represents the final optimal cluster centers. In this work, maximum computing step number is regarded as the halting condition.

Each evolution membrane has the communication rules of the following two forms:

- Rule $[Z_{i,best}]_i \rightarrow Z_{i,best}$, where $Z_{i,best}$ represents the best object in ith evolution membrane after objects are evolved, $i = 1, 2, \ldots, q$.

  In each computing step, we can obtain an object with smallest fitness value in an evolution membrane, for example, $Z_{i,best}$ for the ith evolution membrane. The rule expresses that the best object in the ith evolution membrane is communicated into the skin membrane to update the global best object in the skin membrane. The updating strategy used are described as follows:

$$Z_{best} = \begin{cases} Z_{i,best}, & \text{if } f(Z_{i,best}) < f(Z_{best}) \\ Z_{best}, & \text{otherwise} \end{cases} \tag{5}$$

- Rule $Z_{best} \rightarrow [Z_{best}]_i$, $i = 1, 2, \ldots, q$. The rule indicates that the global best object stored in the skin membrane will enter the ith evolution membrane. The communication rule is applied before objects are evolved and the global best object communicated will participate in the evolution of the objects in each evolution membrane (see Eq.(2)).

The arrows in Fig. 1 indicate the communication relations of objects between elementary membranes and skin the membrane.

2.5. The evaluation of the objects

In partitional clustering, a commonly used metric is employed to evaluate the clustering quality of a clustering algorithm. The clustering metric is defined as the sum of the Euclidean distances of the data points to their corresponding cluster centers, i.e.,

$$M(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \parallel x_j - z_i \parallel \tag{6}$$

where x_j $(j = 1, 2, \ldots, m)$ are data points in D, C_i $(i = 1, 2, \ldots, k)$ are k clusters and z_i $(i = 1, 2, \ldots, k)$ are the corresponding cluster centers. Generally, the smaller the M value is, the higher the clustering quality.

In the cell-like P system, each object in elementary membranes expresses a group of candidate cluster centers, which has a corresponding M value as its fitness value. The goal of the cell-like P system is to search for the optimal cluster centers that have the smallest M value. Therefore, the M value is used as the fitness function to evaluate each object in elementary membranes, namely, the smaller the M value is, the better the object.

2.6. The DE-MC algorithm

The designed cell-like P system is used to find the optimal cluster centers for a data set. The proposed DE-MC algorithm is described in Table 1.

Table 1. The DE-MC clustering algorithm

Input parameters:
  Data set, D;
  Number of elementary membranes, q;
  Number of objects in each evolution membrane, n;
  Maximum computing step number, S_{max};
  Scaling factor, F;
  Crossover rate, CR;
Output results:
  Optimal object, Z_{best};
Step 1. **Initialization**
  for i=1 *to* q
    for j=1 *to* n
      Generate initial jth object $Z_{i,j}$ for ith membrane;
      Calculate the fitness value of $Z_{i,j}$ according to Eq. (6);
    end for
  end for
  Set computing step $s = 0$;
Step 2. **Object evolution in evolution membranes**
  Communicate the global best object into evolution membranes;
  for each elementary membrane i in parallel *do*
    for j=1 *to* n
      Generate $U_{i,j}$ by $Z_{i,j}$ according to the mutation rule;
      Generate $V_{i,j}$ by $Z_{i,j}$ and $U_{i,j}$ according to the crossover rule;
      Generate $Z'_{i,j}$ by $Z_{i,j}$ and $V_{i,j}$ according to the selection rule;
    end for
    Communicate $Z_{i,best}$ into the skin membrane;
    Update the global best object in the skin membrane;
  end for
  Replace $Z_{i,j}$ by $Z'_{i,j}$, $i = 1, 2, \ldots, q, j = 1, 2, \ldots, n$;
Step 3. **Halt condition judgment**
  If $s > S_{max}$ is satisfied then
    Export the global best object in skin membrane, Z_{best};
    HALT;
  else
    $s = s + 1$;
    goto Step 2;
  end if

3. Experiment results and analysis

The proposed DE-MC clustering algorithm has been evaluated on a real-life data set and an artificial data set, and has been further compared with a classical k-means algorithm and two recently developed partitional clustering algorithms, which are genetic algorithm (GA)-based clustering algorithm and differential evolution (DE)-based clustering algorithm, respectively.

3.1. Data sets

In the experiments, a real-life data set (*Iris*) and an artificial data set (*Data1*) are used to evaluate the proposed DE-MC clustering algorithm and other three partitional clustering algorithms mentioned above. The two data sets are described as follows.

- *Iris.* The data set represents different categories of irises, which have four features. The four features represent the sepal length, sepal width, petal length and the petal width in centimeters respectively. The data set has three kinds of irises (there are some overlaps between classes 2 and 3), each of which has 50 samples. So for the data set, the number of the cluster centers k is chosen to be 3. Therefore, each object in elementary membranes is a 12-dimensional vector.

- *Data1.* The data set has nine overlapping classes and all the classes are assumed to have an equal priori probability (=1/9). It has 900 two-dimensional data points, which obey the triangular distribution. The $X - Y$ ranges for the nine classes are given as follows:
  class 1: [-3.3, -0.7] × [0.7, 3.3]
  class 2: [-1.3, 1.3] × [0.7, 3.3]
  class 3: [0.7, 3.3] × [0.7, 3.3]
  class 4: [-3.3, -0.7]× [-1.3, 1.3]
  class 5: [-1.3, 1.3] × [-1.3, 1.3]
  class 6: [0.7, 3.3]× [-1.3, 1.3]
  class 7: [-3.3, -0.7] × [-3.3, -0.7]
  class 8: [-1.3, 1.3]× [-3.3, -0.7]
  class 9: [0.7, 3.3] × [-3.3, -0.7]

  Thus the domain of the triangular distribution for each class and for each axis is 2.6. Consequently, the height will be 1/1.3 (since $2.6 * height/2 = 1$). This data set is shown in Fig. 4. The value of k is chosen to be 9 for this data set. Thus, each object in elementary membranes is a 18-dimensional vector.

3.2. Setup

In the experiments, the parameters of the proposed DE-MC clustering algorithm and of the other three clustering algorithms are set as follows:

- DE-MC clustering algorithm. The number of the elementary membranes is chosen to be 4, the number of objects in each evolution membrane is chosen

to be 100 and the maximum iteration times is 100. In the used differential evolution mechanism, the crossover factor is $CR = 0.3$, and the mutation factor is computed as follows:

$$F = 1 - \sqrt[3]{\frac{f(x_{best})}{f(V_i)}} \tag{7}$$

- GA-based clustering algorithm. In the algorithm, GA uses the known rotating wheel method, single-point crossover and single-point mutation as select, crossover and mutation operations respectively, where the crossover probability and mutation probability are set to be 0.8 and 0.001 respectively. The population size is set to be 100, and the number of the maximum iteration times is 100.
- DE-based clustering algorithm. The population size is set to be 100, and the number of the maximum iteration times is 100. The DE uses the mutation operation of "(DE/best/1/bin)" type, where the crossover factor is set to be CR=0.3, and the mutation factor is given by Eq. (5).
- K-means algorithm. In experiments, the initial clustering centers for the K-means algorithm are generated randomly.

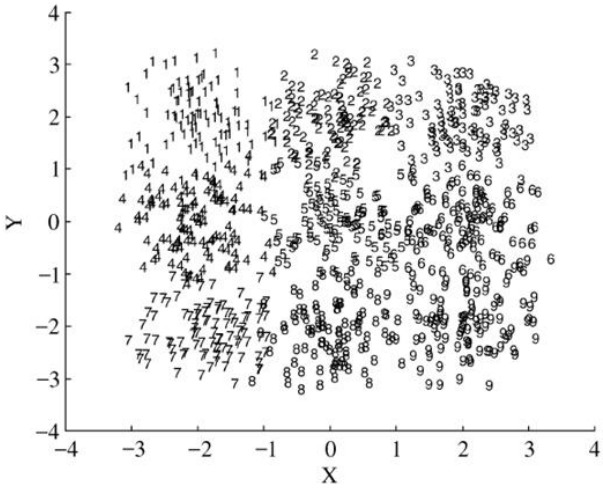

Fig. 4. Artificial data set *Data1*: "1" - points from class 1, "2" - points from class 2, and so on.

3.3. Experimental results

In this work, the M value is used to measure clustering qualities of all clustering algorithms. Considering that the designed cell-like P system and other three clustering algorithms include stochastic mechanisms, we independently execute the five clustering algorithms 5 times on the two data sets, respectively. In addition to the

M value given in each run, we also provide the mean and standard deviation of the five runs.

Table 2. The comparison results of M values for the DE-MC clustering algorithm, GA-based clustering algorithm, DE-based clustering algorithm and k-means on *Iris*

	K-means	GA	DE	DE-MC
run=1	97.325924	96.927123	96.732838	96.700209
run=2	97.346220	96.946688	96.767323	96.664124
run=3	97.346220	97.046805	96.798520	96.682223
run=4	97.346220	96.865863	96.817824	96.667066
run=5	97.325924	97.032286	96.751325	96.683836
Mean	97.338102	96.963753	96.773566	96.679492
Standard deviation	0.011117	0.075515	0.034547	0.014544

Table 2 shows the experiment results of the four clustering algorithms on *Iris* executed 5 times each. The results reveal that the clustering quality of the proposed DE-MC clustering algorithm is better than that of other three clustering algorithms except for the second run. Meanwhile, the average clustering quality of the proposed DE-MC clustering algorithm for 5 runs is superior to that of k-means and GA-based clustering algorithm and DE-based clustering algorithm. The standard deviation of the proposed DE-MC clustering algorithm is slightly higher than that of k-means algorithm, but is better than that of GA-based clustering algorithm and DE-based clustering algorithm. The comparison results indicate that the proposed DE-MC clustering algorithm has a better clustering quality and better stability on *Iris*.

Table 3. The comparison results of M values for the DE-MC clustering algorithm, GA-based clustering algorithm, DE-based clustering algorithm and k-means on *Data1*

	K-means	GA	DE	DE-MC
run=1	651.808891	600.007424	600.290359	599.954833
run=2	655.462382	600.154046	600.097933	600.115086
run=3	600.561969	600.278141	599.995098	599.870488
run=4	702.987431	600.227584	599.848165	599.810946
run=5	648.299526	599.914131	599.998067	599.808638
Mean	651.82404	600.116265	600.045924	599.911906
Standard Deviation	36.301351	0.152256	0.163122	0.128251

The experimental results of the proposed DE-MC, k-means, GA-based and DE-based algorithms for $Data1$ are provided in Table 3. It is clear from Table 3 that, (i) the clustering quality of the proposed DE-MC clustering algorithm is better than

that of other three clustering algorithms, (ii) the average clustering quality of the proposed DE-MC clustering algorithm for 5 runs is superior to that of other three clustering algorithms, and (iii) the standard deviation of the proposed DE-MC clustering algorithm is lower than that of the other three clustering algorithms. The comparison results illustrate that the proposed DE-MC clustering algorithm has a better clustering quality and better stability on *Data1*.

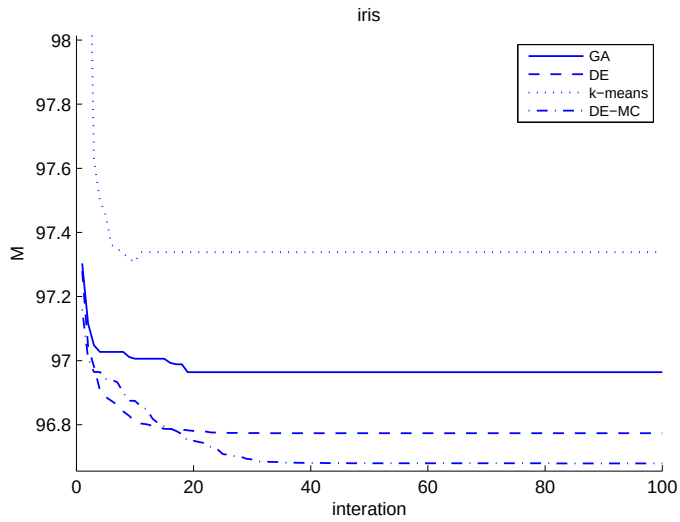

Fig. 5. The average trends of the proposed DE-MC, k-means, GA-based and DE-based algorithms for *Iris*.

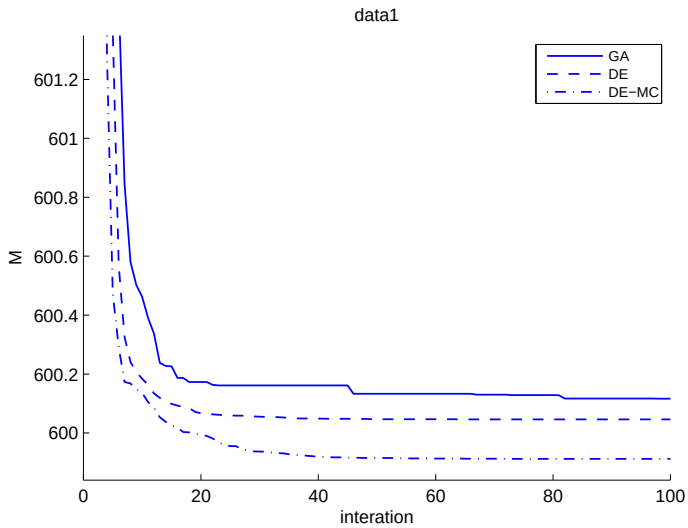

Fig. 6. The average trends of the proposed DE-MC, k-means, GA-based and DE-based algorithms for *Data1*.

Figures 5 and 6 show the average trends for the proposed DE-MC, k-means, GA-based and DE-based algorithms on $Iris$ and $Data1$, respectively. From the figures, it can be found that the proposed DE-MC clustering algorithm is obviously superior to the other three clustering algorithms in terms of convergence speed and clustering quality. The clustering advantage of the proposed DE-MC clustering algorithm benefits from the inherent co-evolution mechanism of multiple elementary membranes in the used cell-like P system.

4. Conclusions

This paper proposed a novel membrane clustering algorithm, which was based on a cell-like P system of a two-layer membranes structure. Each object in membranes represents a group of candidate cluster centers. The differential evolution mechanism of differential evolution algorithm was introduced as evolution rules. Under the control of differential evolution mechanism in multiple elementary membranes, the cell-like P system automatically searched for the optimal cluster centers to achieve data clustering. The proposed DE-MC clustering algorithm has been evaluated on a real-life data set and on an artificial data set, and compared with classical k-means algorithm, GA-based clustering algorithm and DE-based clustering algorithm. The comparison results have indicated that the proposed DE-MC clustering algorithm overall is significantly superior to the other three algorithms in terms of clustering quality and has good robustness and convergence features.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (No. 61170030), Chunhui Project Foundation of the Education Department of China (No. Z2012025, No. Z2012031), Research Fund of Sichuan Key Technology Research and Development Program (No. 2013GZX0155), China.

References

[1] EVERITT B., LANDU S., LEESE M., *Cluster Analysis*, Arnold, London (2001).

[2] NALDI M., CAMPELLO R., HRUSCHKA E., CARVALHO A., *Efficiency issues of evolutionary k-means*, Applied Soft Computing, **11**, pp. 1938–1952 (2011).

[3] XU R., WUNSCH D., *Survey of clustering algorithm*, IEEE Trans. Neural Networks, **16**(3), pp. 645–678 (2005).

[4] KRISHNA K., MURTY M., *Genetic k-means clustering algorithm*, IEEE Trans. on System, Man, and Cybernetics, Part B-Cybernetics, **29**(3), pp. 433–439 (1999).

[5] MUALIK U., BANDYOPADHYAY S., *Genetic algorithm-based clustering technique*, Pattern Recognition, **33**, pp. 1455–1465 (2000).

[6] LASZLO M., MUKHERJEE S., *A genetic algorithm that exchanges neighboring centers for k-means clustering*, Pattern Recognition Letters, **28**, pp. 2359–2366 (2007).

[7] SHELOKAR P., JAYARAMAN V., KULKARNI B., *An ant colony approach for clustering*, Analytica Chimica Acta **509**(2), pp. 187–195 (2004).

[8] NIKNAM T., OLAMAIE J., AMIRI B., *A hybid evolutionary algorithm based on ACO and SA for cluster analysis*, Journal of Applied Science, **8**(15), pp. 1695–2702 (2008).

[9] NIKNAM T., FIROUZI B., NAYERIPOUR M., *An efficient hybrid evolutionary algorithm for cluster analysis*, World Applied Science Journal, **4**(2), pp. 300–307 (2008).

[10] PĂUN GH., *Computing with membranes*, Journal of Computer System Sciences, **61**(1), pp. 108–143 (2000).

[11] PĂUN GH., ROZENBERG G., SALOMAA A. eds., The Oxford Handbook of Membrane Computing. Oxford University Press, New York (2010)

[12] IONESCU M., PĂUN GH., YOKOMORI T., *Spiking neural P systems*, Fundamenta Informaticae, **71**(2–3), pp. 279–308 (2006).

[13] WANG J., ZHOU L., PENG H., ZHANG G., *An extended spiking neural P system for fuzzy knowledge representation*, International Journal of Innovative Computing, Information and Control, **7**(7A), pp. 3709–3724 (2011).

[14] FREUND R., PĂUN GH., PÉREZ-JIMÉNEZ M.J., *Tissue-like P systems with channel-states*, Theoretical Computer Science, **330**(1), pp. 101–116 (2005).

[15] PĂUN GH., PÉREZ-JIMÉNEZ M.J., *Membrane computing: brief introduction, recent results and applications*, BioSystems, **85**(1), pp. 11–22 (2006).

[16] PENG H., WANG J., PÉREZ-JIMÉNEZ M.J., WANG H., SHAO J., WANG T., *Fuzzy reasoning spiking neural P system for fault diagnosis*, Information Sciences, **235**, pp. 106–116 (2013).

[17] WANG J., SHI P., PENG H., PÉREZ-JIMÉNEZ M.J., WANG T., *Weighted fuzzy spiking neural P systems*, IEEE Transactions on Fuzzy Systems, **21**(2), pp. 209–220 (2013).

[18] PENG H., WANG J., PÉREZ-JIMÉNEZ M.J., SHI P., *A novel image thresholding method based on membrane computing and fuzzy entropy*, Journal of Intelligent & Fuzzy Systems, **24**(2), pp. 229–237 (2013).

[19] NISHIDA T., *An approximate algorithm for NP-complete optimization problems exploiting P systems*, *Proceedings of the Workshop on Uncertainty in Membrane Computing*, Palma de Mallorca, pp. 185–192 (2004).

[20] ZHANG G., CHENG J., GHEORGHE M., MENG Q., *A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems*, Applied Soft Computing, **13**(3), pp. 1528–1542 (2013).

[21] STORN R., PRICE K., *Differential evolution – a simple and efficient heuristic strategy for global optimization over continuous spaces*, Journal of Global Optimization, **11**, pp. 341–359 (1997).

[22] PRICE K., STORN R., LAMPINEN J., *Differential Evolution – A Practical Approach to Global Optimization*, Springer, Berlin (2005).