Research Article

A Complex Chained P System Based on Evolutionary Mechanism for Image Segmentation

Xiyu Liu,1,2 Lin Wang,1,2 Jianhua Qu,1,2 and Ning Wang1,2

1Institute of Management Science, Shandong Normal University, Jinan 250000, China
2College of Business, Shandong Normal University, Jinan, China

Correspondence should be addressed to Xiyu Liu; xyliu@sdnu.edu.cn

Received 1 December 2019; Revised 6 February 2020; Accepted 25 February 2020; Published 7 August 2020

Academic Editor: Rodolfo E. Haber

Copyright © 2020 Xiyu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new clustering membrane system using a complex chained P system (CCP) based on evolutionary mechanism is designed, developed, implemented, and tested. The purpose of CCP is to solve clustering problems. In CCP, two kinds of evolution rules in different chained membranes are used to enhance the global search ability. The first kind of evolution rules using traditional and modified particle swarm optimization (PSO) clustering techniques are used to evolve the objects. Another based on differential evolution (DE) is introduced to further improve the global search ability. The communication rules are adopted to accelerate the convergence and avoid prematurity. Under the control of evolution-communication mechanism, the CCP can effectively search for the optimal partitioning and improve the clustering performance with the help of the distributed parallel computing model. This proposed CCP is compared with four existing PSO clustering approaches on eight real-life datasets to verify the validity. The computational results on tested images also clearly show the effectiveness of CCP in solving image segmentation problems.

1. Introduction

Membrane computing, also known as membrane systems or P systems, is a novel research of bioinspired computing initiated by Păun [1]. It seeks to discover novel biological computing models from the structure of biological cells as well as the cooperation of cells in tissues, organs, and populations of cells. Over the past years, there are three investigated P systems, cell-like P systems, tissue-like P systems, and neural-like P systems, including spiking neural P systems.

P systems have several characteristics: nondeterminism, programmability, extensibility, and readability [2]. Research shows that the some models of P systems present the same computing power as Turing machines and is more efficient to some extent [3]. Therefore, the analysis of computing power and computational efficiency of P systems is one of the important basic studies [4, 5]. Other studies are focused on the variation of P systems to solve optimization problems, including the variant of rules and structures [6, 7]. In addition, some intelligence techniques, such as evolutionary computation and fuzzy theory, is also introduced to the variant P systems in some specific optimization problems [8].

Because the parallel computation in membrane systems can avoid the increase in time consumption with the increase in the number of data points, the membrane systems are suitable for solving clustering problems [9]. There are a lot of interesting researches in variant P systems to solve clustering problems. Liu and Xue [10] proposed a new cluster splitting technique based on Hopfield networks and P systems. Liu et al. [11] presented an improved Apriori algorithm, named ECTPPT-Apriori, based on evolution-communication tissue-like P system with promoters and inhibitors. Peng et al. [12] developed an extended P system with active membranes, in which a modification differential evolution mechanism is used to find the optimal clustering centers in clustering problems. Peng et al. [13] introduced a multiobjective clustering framework using a tissue-like P system for solve fuzzy clustering problems. Wang et al. [14] proposed a new cell-like P system using a modified genetic algorithm to evolve the objects and using communication rules in the cell-like P system.
Image segmentation is an important part of image processing; it also has critical impact on the final quality of image analysis and subsequent tasks [15]. In the previous studies, the segmentation technique can be divided into region-based methods, edge-based methods, cluster-based methods, and threshold-based methods [16], in which the threshold methods can be classified into bilevel and multilevel threshold methods based on the number of clusters [17]. The region-based methods can obtain high segmentation quality but are sensitive to the parameters. The edge-based methods have high segmentation quality in different regions or targets and are also more sensitive to noise. The cluster-based methods are simple and easily implemented, but the clustering results rely on the number of clusters and feature selection in the colour space. The threshold-based methods are simple to computation, requiring no prior knowledge, but the continuity of the regions is not guaranteed due to the lack of space information [18]. So far, image segmentation has been used with wide recognized significance in machine vision, computer-aided diagnosis of the medical imaging, feature extraction, and analysis [19]. In this paper, the experiment on tested images is simple and easily implemented, and the application of image segmentation is not used in our works, so more details are not described in the following.

Because these segmentation methods mentioned above have their respective advantages and limitations, a lot of works have been done to find robust and optimum segmentation techniques [20]. The threshold technique is one of the most popular segmentation techniques which are based on the gray level of images. It is simplicity and easy implementation, which has lower computational complexity [21]. Li et al. [22] presented a novel thresholding extraction method based on variational mode decomposition (VMD). Zhao et al. [23] introduced a gradient-based adaptive particle swarm optimization (PSO) combined with improved external optimization (EO) to overcome being trapped into local optima in high-dimensional tasks. Wang et al. [24] designed a new P system with related interaction rules; the PSO mechanism is used to maximize entropy threshold. Tan et al. [25] proposed a hybrid clustering models using ensemble deep neural networks for skin lesion segmentation.

P systems are a class of distributed parallel computing models that can be used to improve the global search ability of PSO [26]. The commutation rules of chained P systems can be used to accelerate convergence of PSO. Compared with the crossover and mutation operation of DE, the partitioning information only can be used in the velocity updating of particles [27, 28]. Besides, although the neural networks have high quality for solving the optimization problems, the space and time consumption is too much, and the parallel computing model of P systems is also not executed on the neural networks. Therefore, a new variant of P systems based on PSO mechanism is proposed, which is named complex chained P (CCP) systems. And, the concepts of membrane, objects, and rules based on a special chained structure are introduced in the literature [29]. In this CCP system, two kinds of evolution rules in different chained membranes are introduced to enhance the global search ability. One of evolution rules is using the traditional and modified PSO mechanism to evolve the objects, and the partitioning information is introduced as the environmental factor to improve the clustering performance. Another is based on differential evolution (DE) to evolve the global chained objects in order to enhance the global search ability. Compared with genetic algorithm (GA) [30], the DE is simple and easily implemented, which has less predefined parameters and quick convergence speed. The communication rules for global objects between chained membranes are used to accelerate the convergence speed and avoid premature [31, 32]. At last, the CCP system is evaluated on eight benchmark clustering problems and eight tested images with the compared clustering and image segmentation techniques; the experimental results verify the validity and performance of proposed CCP.

The rest of this paper is organized as follows: the framework of the chained P systems is described in Section 2. Section 3 gives more details about the complex chained P system for clustering problems, and the evolution rules and communication rules are described in this section. In order to verify the validity of CCP, some experiments which are conducted on benchmark clustering problems, are taken in Section 4.2. Furthermore, some tested images are used to evaluate the competitive performance of CCP in Section 4.3. Section 5 provides some conclusions and outlines future research directions.

2. Chained P Systems

Some concepts can be briefly defined as follows. A special membrane is called chained membrane, which contains many objects based on a chained structure, and these objects are called chained object. The formal descriptions of the chained P system can be expressed as the following tuple [29].

\[ \Pi = (O, \mu, w_1, \cdots, w_m, R_1, \cdots, R_m, \sigma_{\text{in}}, \sigma_{\text{out}}), \]

where

(1) \( O \) is a finite nonempty alphabet, and the symbols are called chained objects.
(2) \( \mu \) is the membrane structure consisting of \( m \) membranes, which is composed of two parts: the structure of chained membranes and the structure of the whole chained P system.
(3) \( w_1, \cdots, w_m \) are the initial finite sets of chained objects; \( w_i \) represents the chained objects in membrane \( i \), for \( 1 \leq i \leq m \), \( w_i \in O \).
(4) \( R_1, \cdots, R_m \) are the finite sets of chained rules, and \( R_i \) represents a finite set of chained rules in membrane \( i \), for \( 1 \leq i \leq m \), \( R_i = \{R_{i1}, R_{i2}, \cdots, R_{ir}\} \) which consists of \( r \) subrules that are executed on a special order. There are many chained rules on objects, for example, object addition rule, object subtraction rule, object crossover rule, and object variation rule.
3. Proposed Complex Chained P System for Clustering

The proposed chained P system based on evolutionary mechanism with a complex chained structure for clustering problems is presented in this section. First, the general framework of this complex chained P system (CCP) is described, and the basic membrane structure is given. Next, the evolution and communication mechanisms of chained P system are introduced in the CCP system. Finally, the computational process of the CCP system is described in the last parts.

3.1. General Framework of CCP System. The general framework of CCP is similar to that of the chained P systems, but the main differences are the membrane structure of whole P system and the evolution rules for chained objects. In complex chained P system, there are two kinds of chained membranes, chained membranes \( \sigma_1 \) to \( \sigma_m \) and membranes \( \sigma_{m+1} \), which contains different evolution rules. These chained membranes are labelled from 1 to \( m + 1 \) respectively. The formal descriptions of the CCP system are as follows:

\[
\Pi = \langle O, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m, R_1', \ldots, R_m', R_{m+1}', R_{m+1}, \sigma_{in}, \sigma_{out} \rangle,
\]

where

1. \( O \) is a finite set of alphabets, which includes all chained objects or strings in the CCP systems.
2. \( \mu \) is the membrane structure of CCP system consisting of \( m + 3 \) membranes.
3. \( w_1, \ldots, w_m \) are the multisets of the initial chained objects, with \( w_i \in O \), for \( 1 \leq i \leq m \), and \( w_0 \) represents the chained objects.
4. \( R_1, \ldots, R_m \) represent finite sets of evolution rules in chained membranes \( \sigma_1 \) to \( \sigma_m \), \( \sigma_{m+1} \), which represent a set of evolution subrules in chained membrane \( i \), and the number of the subrules are denoted by \( r_i \). The subrule evolution rules are of the form: \( R_i = \{ u \rightarrow v \} \), with \( u, v \in O \), which means that chained object \( u \) will be evolved to chained object \( v \).
5. \( R_1', \ldots, R_{m+1}' \) represent the communication rules in chained membranes \( \sigma_1 \) to \( \sigma_{m+1} \). The communication rules are of the form: \( R_i = \{ u \rightarrow (v, i) \} \), with \( u, v \in O \), for \( 1 \leq i \leq m + 1 \), which means the chained object \( u \) in membrane \( i \) will be changed in object \( v \) and transported into membrane \( j \).
6. \( R_{m+1}' \) represents finite sets of evolution rules in chained membrane \( \sigma_{m+1} \). For two arbitrary chained objects \( u \) and \( v \), if \( u, v \in O \), there are two kinds of evolution rules existing in the membranes can be executed: crossover rules and variation rules, \( R_{m+1}' = \{ R_{m+1}'_1, R_{m+1}'_2 \} \).
7. \( \sigma_{in} \) is the input region or membrane in the CCP system, which contains the initial objects of the whole system.
8. \( \sigma_{out} \) is the output region or membrane in the CCP systems. Once the computation is completed, the computation results or objects will be transported to the output region or membranes. Figure 1 gives the membrane structure of the proposed CCP system.

In this proposed CCP, the chained objects are considered to be feasible solutions in the search space, and the partition obtained by the clustering technique is represented by \( C = \{ C_k \} \), for \( 1 \leq k \leq K \). Thus, the \( i \)-th chained object \( u_i \) can be defined as \( u_i = (z_{i1}, z_{i2}, \ldots, z_{ik}) \), where \( z_{ik} \), \( 1 \leq k \leq K \) represents the \( k \)-th cluster center, \( K \) is the number of the clusters, and the dimension of cluster center is denoted by \( d \).

At last, the best chained object that represents the best partitioning results in the system will be output to the \( \sigma_{out} \) when the computation is completed.

3.2. Evolution Rules on Chained Objects. There are two kinds of evolution rules on chained objects existing in the chained membranes, the evolution rules in chained membranes \( \sigma_1 \) to \( \sigma_m \) and another evolution rules in chained membrane \( \sigma_{m+1} \).

3.2.1. Evolution Rules in Chained Memories. There are two kinds of evolution subrules in chained membranes \( \sigma_1 \) to \( \sigma_m \), which are based on different velocity updating strategies for chained objects. Firstly, the particle swarm optimization (PSO) mechanism is used to evolve the chained objects, and the traditional velocity model of PSO is introduced to update the velocity of chained object. At \( t + 1 \) time, the velocity formula of the chained object \( u_i \) in \( j \)-th chained membrane \( \sigma_{j} \), \( 1 \leq j \leq m \) is determined by (3) in the following equation:

\[
V_i(t+1) = wV_i(t) + c_1r_1(u_i^{best}(t) - u_i(t)) + c_2r_2(u_j^{best}(t) - u_i(t)),
\]

where the inertia weight is denoted by \( w \) and \( c_1 \) and \( c_2 \) represent the learning factor, which is usually restricted to 2. The iteration counter is denoted by \( t \), \( r_1 \) and \( r_2 \) are two independent uniform random numbers. The local best of \( u_i \) in history is denoted by \( u_i^{best} \), specially, the global best in \( \sigma_j \) is denoted by \( u_j^{best} \).

Another velocity updating strategy for objects is based on environmental factors. At \( t + 1 \) time, a modified velocity formula of EPSO is determined by (4) in the following equation:
\[ V_i(t + 1) = u V_i(t) + c_1 r_1 (u_i^{best}(t) - u_i(t)) + c_2 r_2 (u_j^{best}(t) - u_i(t)) + c_3 r_3 (E_i(t) - u_i(t)), \]

where \( c_1 \) represents a positive constant, \( r_1 \) is a uniform random number, and \( E_i(t) \) represents the environmental factor around to the object \( u_i \), which is based on the information of clusters. The partitioning information of cluster is changed dynamically through the evolution of objects, and the geometric center of the data point belonging to the corresponding cluster in the object \( u_i \) is used as the environment factor, where \( E_i(t) = \{ e_{i1}(t), e_{i2}(t), \ldots, e_{ik}(t) \} \), for \( k = 1, 2, \ldots, K \), which is given by (5) at \( t \) time in the following equation:

\[ e_{ik}(t) = \frac{\sum_{x_p \in C_i} x_p}{N_k}, \]

where \( e_{ik}(t) \) represents the \( k \)-th cluster centroid of the environment factor \( E_i \), \( x_p \) represents the \( p \)-th data point in the dataset, for \( p = 1, 2, \ldots, N \), \( N \) is the number of the data point in the datasets, and number of data points belonging to the corresponding cluster \( C_k \) is denoted by \( N_k \).

At \( t + 1 \) time, the position formula of \( u_i \) is determined by (6) in the following equation:

\[ u_i(t + 1) = u_i(t) + V_i(t + 1), \]

and the local best of \( u_i \) at \( t + 1 \) time is updated according to (7) in the following equation:

\[ u_i^{best}(t + 1) = \begin{cases} u_i(t + 1), & \text{if } f(u_i(t + 1)) < f(u_i^{best}(t)), \\ u_i^{best}(t), & \text{otherwise}, \end{cases} \]

where \( f(u) \) represents the fitness function of \( u_i \) for clustering problems, which can be defined by (8) in the following equation:

\[ f(u) = \min \sum_{k=1}^{K} \sum_{j=1}^{N} \| x_j - z_{ik} \|. \]

The purpose of clustering problems is to find a partitioning result \( C = \{ C_1, C_2, \ldots, C_K \} \) obtained by techniques to minimize the values of fitness function according to equation (8). The global best \( u_j^{gbest} \) at \( t + 1 \) time is updated according to (7) in the following equation:

\[ u_j^{gbest}(t + 1) = \begin{cases} u_j^{gbest}(t + 1), & \text{if } f(u_j^{gbest}(t + 1)) < f(u_j^{gbest}(t)), \\ u_j^{gbest}(t), & \text{otherwise}. \end{cases} \]

At \( t \) time, a success rate of chained objects \( u_i \) is defined by the following equation:

\[ S(u_i, t) = \begin{cases} 1, & \text{if } f(u_i^{gbest}(t)) < f(u_i^{gbest}(t - 1)), \\ 0, & \text{otherwise}, \end{cases} \]

where \( S(u_i, t) \) represents the success rate of \( u_i \). The success rate in the chained membrane \( \sigma_j \), for \( 1 \leq j \leq m \), is computed by (11) in the following equation:

\[ P(\sigma_j, t) = \frac{\sum_{i=1}^{n_j} s(u_i, t)}{n_j}, \]

where \( n_j \) represents the number of the chained object in chained membrane \( \sigma_j \) and \( P(\sigma_j, t) \) is the success rate of \( \sigma_j \), \( P(\sigma_j, t) \in [0, 1] \), which means the improvement rate at last computation. In this study, a linearly increasing strategy is introduced to adjust the values of the inertia weight dynamically; the possible range is based on the mapping relationship of \( P(\sigma_j, t) \), which is given by (12) in the following equation:

\[ w(t) = (w_{max} - w_{min})P(\sigma_j, t) + w_{min}, \]
where $w_{\text{min}}$ and $w_{\text{max}}$ represent the minimum and maximum of inertia weight and $\omega(t) \in [w_{\text{min}}, w_{\text{max}}]$.

### 3.2.2. Evolution Rules in Chained Membrane $\sigma_{m+1}$

The evolution rules in chained membrane $\sigma_{m+1}$ contain objects crossover rules and object variation rules, differential evolutionary (DE) approach is further used as the variant of evolution rules in $\sigma_{m+1}$, and the mutation and crossover mechanism is also introduced to help the objects escape the local optima. At $t$ time, the $i$-th chained object in $\sigma_{m+1}$ is denoted by $u_i(t) = (u_{i1}(t), u_{i2}(t), \ldots, u_{iD}(t))$, where $D = k \times d$ represents the dimension of the objects and $u_i(t)$ and $u_j(t)$ are two randomly chained objects in $\sigma_{m+1}$; the mutation operation is defined by (13) in the following equation:

$$v_{iy}(t) = u_{iy}(t) + F_1 \times (u_{\text{gbest}}(t) - u_{iy}(t)) + F_2 \times (u_{iy}(t) - u_{iy-1}(t)), \quad \text{for } y = 1, 2, \ldots, D,$$

(13)

where $v_{iy}(t)$ represents the created donor, $u_{\text{gbest}}(t)$ represents the best object in $\sigma_{m+1}$, the scaling factor is denoted by $F_1$ and the value is given by $F_1 = 0.5 \times (1 + \text{rand}(0, 1))$, $F_2 = 0.5 \times (1 + \text{rand}(0, 1))$. The mutant $u_i'(t)$ comes from the created donor through crossover mechanism, which can be defined by

$$u_i'(t) = \begin{cases} v_{iy}(t), & \text{if } \text{rand}_i \leq P_c \text{ or } y = \text{rand}_y, \\ u_{iy}(t), & \text{otherwise}, \end{cases} \quad \text{for } y = 1, 2, \ldots, D,$$

(14)

where the crossover rate is denoted by $P_c$, it is a predefined constant within the range from 0 to 1; rand, is an independent uniform random number, and rand is the random dimension from 1 to $D$. At $t + 1$ time, the position formula of $u_i$ is determined by

$$u_i(t + 1) = \begin{cases} u_i'(t), & f(u_i'(t)) < f(v_i(t)), \\ u_i(t), & \text{otherwise}. \end{cases}$$

(15)

### 3.3. Communication Rules in Chained Membranes

The communication rules of chained P system are used to enhance the cooperation between the chained membranes, which provide good foundation for exchange and sharing of the information. There are two kinds of communication rules in the CCP system:

1. $R^j_{m+1} = \left\{ u_{\text{gbest}}^{\text{out}}(t) \rightarrow (u_{j}(t), \text{in}_{m+1}) \right\}, \quad \text{for } j = 1, 2, \ldots, m$. At $t$ time, the global best $u_{\text{gbest}}^{\text{out}}$ in $\sigma_{j}$ is sent to the chained membrane $\sigma_{m+1}$ and transformed into the $j$-th object $u_{j}$. At each iterative computation, the chained membrane $\sigma_{m+1}$ only contains $m$ objects that come from the chained membranes $\sigma_{1}$ to $\sigma_{m}$.

2. $R^m_{m+1} = \left\{ u_{\text{gbest}}^{\text{out}}(t) \rightarrow (u_{\text{gbest}}^{\text{out}}(t), \text{in}) \right\}, \quad \text{for } j = 1, 2, \ldots, m$. At $t$ time, the global best $u_{\text{gbest}}^{\text{out}}$ in $\sigma_{m+1}$ is transported to the chained membranes $\sigma_{1}$ to $\sigma_{m}$, which is considered to be the global best $u_{\text{gbest}}^{\text{out}}$ in chained membrane $\sigma_{j}$. Meanwhile, the global best $u_{\text{gbest}}^{\text{out}}$ in membrane $\sigma_{m+1}$ is sent to the output membrane $\sigma_{\text{out}}$, which is viewed as the best object or the final computation results of the CCP at $t$ time.

### 3.4. Computation Process of CCP

#### 3.4.1. Initialization

The input membrane $\sigma_{\text{in}}$ contains all initial objects in the P system, denoted by $Q = m \times n$. The position of chained object is randomly initialized in the search space. After initialization, the chained objects in the membrane $\sigma_{\text{in}}$ will be transported to the chained membrane $\sigma_{1}$ to $\sigma_{m}$, and each chained membrane contains $n$ objects.

#### 3.4.2. Evolution in Chained Membranes $\sigma_{1}$ to $\sigma_{m}$

The evolution rules on chained objects are used to completed the evolution process according to the equations (3), (4), and (6) in chained membrane $\sigma_{j}$, $(1 \leq j \leq m)$. And, the selection for velocity formula is based on a random strategy. The local best and global best in $\sigma_{j}$ are updated through the equations (7) and (9).

#### 3.4.3. Communication between Chained Membranes

The first kinds of communication rules are used to transport the global best in the $\sigma_{j}$, $(1 \leq j \leq m)$ to the chained membrane $\sigma_{m}$.

#### 3.4.4. Evolution Rules in Chained Membranes $\sigma_{m+1}$

The evolution rules on chained objects are used to completed the evolution process according to the equations (13) and (14) in chained membrane $\sigma_{m+1}$, and global best in $\sigma_{m+1}$ is updated by the equation (15).

#### 3.4.5. Communication between Chained Membranes

The second kinds of communication rules are used to transport the global best in $\sigma_{m}$ to each chained membrane $\sigma_{j}$, $(1 \leq j \leq m)$.

#### 3.4.6. Halting and Output

The evolution and communication will be implemented repeatedly with an iterative form during the computation process. The termination criterion of the system is stopped to whether the maximum number of iterations or computation is reached. When the system halts, the output membrane $\sigma_{\text{out}}$ will send the global best to the environment, and this object is regarded as the final computing results of CCP system.

### 4. Experimental Analysis of Clustering Problems

In this section, the feasibility and effectiveness of the proposed CCP will be demonstrated through the experimental analysis. The datasets of the experiment are introduced first, and the artificial dataset from the previous studies [33] is used to tune the parameters in CCP. Eight real-life datasets from UCI machine leaning repository [34] is used to
compare the performance with currently existing clustering approaches. All clustering approaches, including CCP, are implemented on MATLAB 2016b, and all the experiments are conducted on a DELL desktop computer with an Intel 4.00GHz i7-8550U processor and 8 GB of RAM in a Windows 10 environment.

4.1. Parameter Setting. The numbers of chained membranes have important influences on the performance of CCP. Therefore, four artificial datasets [33], Data_5_2, Data_9_2, Size_5, and Square4, are used to tune this parameter in order to ensure the equity of the experiment. The details of the four artificial datasets are given in Table 1.

Different CCP systems with different degrees of \( m \) = 4, \( m = 5 \), \( m = 6 \) [12] are used to evaluate the effects of the number of the chained membranes. The maximum of iterations is set to \( t_{\text{max}} = 100 \), and the positive constant \( c_3 \) is a random number which distributed to 0.6 and 3. The lower and upper limits of the inertia weight are set to \( w_{\text{min}} = 0.4 \) and \( w_{\text{max}} = 1.2 \). The mutation probability \( P_e \) is randomly generated from 0.2 to 0.8. Other parameters which are not tested in this experiment will be maintained the same values for the fairness of the comparison experiments. And, the number of independent running is set to 30 to eliminate the effect of the random factors. The values of mean and SD of the fitness function obtained by CCP system with different degrees are reported in Table 2.

Table 2 reports the clustering results of the CCP system with different degrees on four artificial datasets. The best values of mean and SD for each dataset are given in bold. These results show that the mean and the SD values obtained by the CCP system when \( m = 10 \) are the best among these different systems.

4.2. Clustering Problems. The performance of the CCP is compared with four clustering approaches that have been reported in the literature to further evaluate the effectiveness, such as standard particle swarm optimization (PSO) [35], differential evolution (DE) [36], environment particle swarm optimization (EPSO) [37], and adaptive particle swarm optimization (APSO) [38]. The comparison experiments are conducted on eight real-life datasets from the UCI machine learning repository, Iris, Newthyroid, Seeds, Diabetes, Yeast, Glass, CMC, and Lung Cancer. More details about these datasets are presented in Table 3.

In these compared clustering approaches, the crossover probability \( P_e \) of the DE approach is randomly generated from 0.2 to 0.8. An environment factor based on the clustering information is embedded in the velocity updating model of EPSO approach, and the learning factor \( c_3 \) is randomly generated from 0.6 to 3. A nonlinear regressive function of APSO based on the population diversity is used to the adjustment formula of the inertia weight, \( L \) represents a predefined constant, and \( c \) is a predefined constant. All adjustable parameters in these clustering approaches are set to the appropriate values which are reported in the respective publications from Table 4.

Each clustering approach, including CCP, ran for 50 times for each dataset to eliminate the effects of random factors. Simple statistics including worst values (Worst), best value (Best), Mean, and SD of fitness function according to (8) are used in the experiments as the evaluation criteria of clustering results. The experimental environment is the same for all comparative clustering approaches.

Figure 2 shows the convergence of these clustering approaches on the eight test datasets for typical runs of these approaches. The fitness value obtained by CCP declines faster at the beginning of the evolution process and then obtains fine convergence for each dataset. The values of the fitness function of PSO and DE decrease slowly at the beginning of the evolution process and do not apparently have better convergence performance than other approaches. Although EPSO and APSO show better performance than the above clustering approaches, they are also easily trapped into local optima, as shown in Figures 2(e), 2(f), and 2(h). Therefore, CCP has better convergence speed and higher clustering quality than the comparative approaches for all these datasets, as shown in Figure 2.

Simple statistics of the fitness function values of these clustering approaches on these datasets are reported in Table 5. Results in Table 5 show that CCP has the overall best performance on these eight test datasets. Due to the characteristics of the test datasets, some clustering approaches performed better on some specific datasets with smaller SD, but the performance of CCP on these ten datasets is all considered comparable. Table 6 provides the average computation time in seconds taken by each of the five clustering techniques when running 50 times on each of the datasets. It can be seen from Table 6 that the proposed CCP has a larger average computation time as compared with PSO, DE, EPSO, and APSO. The evolution process in the chained membranes is time-consuming so that CCP takes more time than other techniques does.

In order to evaluate the clustering performance of these compared clustering approaches, clustering accuracy (CA) is used to evaluate the quality of clustering results obtained by the clustering techniques; the overall accuracy of partitioning results is defined by

\[
\text{CA} = \frac{\sum_{k=1}^{K} \max_{l=1}^{K} |C_k \cap C_l|}{N},
\]

where \( |C_k \cap C_l| \) represents the number of the data points in both belonging to actual cluster \( k \) and partitioning cluster \( l \), for \( k, l = 1, 2, \ldots, K \). The simple statistics of clustering approaches on eight datasets are reported in Table 7. It can be seen that CCP has overall better performance on these datasets. Although some approaches show a better performance on some specific datasets, the performance of CCP on

| Table 1: Properties of the artificial datasets. |
|-----------------------------------------------|
| Datasets | Data | Feature | Class |
|-----------------|-------|--------|-------|
| Data_5_2        | 250   | 2      | 5     |
| Data_9_2        | 900   | 2      | 9     |
| Size_5          | 1000  | 2      | 4     |
| Square4         | 1000  | 2      | 4     |
Table 2: Performance of CCP with different degrees.

| Datasets       | Mean | SD  | Mean | SD  | Mean | SD  |
|----------------|------|-----|------|-----|------|-----|
| Data_5_2       | 326.5071 | 0.0301 | 326.4696 | 0.0123 | 326.4641 | 0.0110 |
| Data_9_2       | 590.8587 | 0.0624 | 590.7589 | 0.0463 | 590.7413 | 0.0287 |
| Size_5         | 2493.1450 | 4.8048 | 2491.9680 | 0.0789 | 2491.9020 | 0.0565 |
| Square4        | 2367.6443 | 0.0317 | 2367.6052 | 0.0260 | 2367.5917 | 0.0141 |

Table 3: Properties of the UCI datasets.

| Datasets        | Data | Feature | Class |
|-----------------|------|---------|-------|
| Iris            | 150  | 4       | 3     |
| Newthyroid      | 215  | 5       | 3     |
| Seeds           | 210  | 7       | 3     |
| Diabetes        | 768  | 8       | 2     |
| Yeast           | 1484 | 8       | 10    |
| Glass           | 214  | 9       | 6     |
| CMC             | 1473 | 9       | 3     |
| Lung Cancer     | 32   | 26      | 3     |

Table 4: Parameter setting of the comparative clustering approaches used in the experiments.

| Parameters      | PSO  | DE   | EPSO | APSO | CCP  |
|-----------------|------|------|------|------|------|
| Population (Q)  | 100  | 100  | 100  | 100  | 100  |
| $t_{\text{max}}$ | 200  | 200  | 200  | 200  | 200  |
| $c_1, c_2$      | 2.2  | N    | 2.2  | N    | 2.2  |
| $c_3$           | N    | N    | (0.6, 3) | N | (0.6, 3) |
| $r_1, r_2$      | (0,1)| N    | (0, 1) | N | (0, 1) |
| $r_3$           | N    | N    | (0, 1) | N | N    |
| $(w_{\text{min}}, w_{\text{max}})$ | 1    | N    | (0.4, 0.6) | N | (0.4, 1.2) |
| $P_c$           | N    | (0.2, 0.8) | N | N | (0.2, 0.8) |
| $L$             | N    | N    | N    | 2.1 | N    |
| $\epsilon$      | N    | N    | N    | 2   | N    |
| $m$             | N    | N    | N    | 10  | N    |

Figure 2: Continued.
Figure 2: Convergence of the approaches on each of the eight real-life datasets. (a) Iris, (b) Newthyroid, (c) Seeds, (d) Diabetes, (e) Yeast, (f) Glass, (g) CMC, and (h) Lung Cancer.
these specific datasets is also considered compared from Table 7.

4.3. Proposed CCP for Image Segmentation. In this section, some typical experiment and analysis for tested image are proposed to evaluate the segmentation performance of this proposed CCP. These tested images are used in the previous studies and researches, which are provided from the Berkeley segmentation dataset and benchmark [39]. The size of the tested image is $481 \times 321$; Figure 3 gives the original image of these tested images.

OSTU proposed by Qtsu [40] is one of most popular segmentation methods, and it has been used to determine whether the optimal threshold method can give a satisfactory segmentation results. The following discriminant criterion measure of OSTU can be described as follows: An image contains $N$ pixels from 0 to $L$, usually $L$ is set to 255, need to segmented in $M + 1$ clusters. Thus, $M$ thresholds $\{t_1, t_2, \cdots, t_M\}$ are needed that divided the original image.
| Datasets  | Statistics | PSO   | DE    | EPSO  | APSO  | CCP   |
|----------|------------|-------|-------|-------|-------|-------|
| Iris     | Worst      | 0.6667| 0.8933| 0.8933| 0.9000| 0.8667|
|          | Best       | 0.9267| 0.9000| 0.9000| 0.9000| 0.9600|
|          | Mean       | 0.8077| 0.8983| 0.8997| 0.9000| 0.9013|
|          | SD         | 0.0904| 0.0027| 0.0015| 1.03E−17| 0.0248|
| Newthyroid| Worst    | 0.6977| 0.7488| 0.7721|| 0.7767|
|          | Best       | 0.8372| 0.8093| 0.8047| 0.8047| 0.8014|
|          | Mean       | 0.7391| 0.7777| 0.8000| 0.8000| 0.8000|
|          | SD         | 0.0337| 0.0187| 0.0114| 0.0114| 0.0080|
| Seeds    | Worst      | 0.6476| 0.8477| 0.8952| 0.8952| 0.8952|
|          | Best       | 0.8953| 0.9143| 0.9143| 0.9143| 0.9143|
|          | Mean       | 0.8367| 0.8814| 0.8952| 0.8952| 0.8952|
|          | SD         | 0.0850| 0.0207| 1.14E−16| 1.14E−16| 1.14E−16|
| Diabetes | Worst      | 0.6510| 0.6510| 0.6510| 0.6510| 0.6510|
|          | Best       | 0.6510| 0.6510| 0.6510| 0.6510| 0.6510|
|          | Mean       | 0.6510| 0.6510| 0.6510| 0.6510| 0.6510|
|          | SD         | 1.14E−16| 1.14E−16| 1.14E−16| 1.14E−16| 1.14E−16|
| Yeast    | Worst      | 0.3120| 0.3309| 0.4508| 0.3942| 0.4683|
|          | Best       | 0.3376| 0.4016| 0.5370| 0.4569| 0.5418|
|          | Mean       | 0.3171| 0.3640| 0.4965| 0.4377| 0.5200|
|          | SD         | 0.0057| 0.0248| 0.0275| 0.0134| 0.0152|
| Glass    | Worst      | 0.3551| 0.4720| 0.5607| 0.5093| 0.5841|
|          | Best       | 0.4813| 0.5234| 0.5981| 0.5841| 0.5841|
|          | Mean       | 0.4140| 0.5021| 0.5834| 0.5486| 0.5841|
|          | SD         | 0.0444| 0.0144| 0.0222| 0.0267| 1.14E−16|
| CMC      | Worst      | 0.4332| 0.4358| 0.4515| 0.4562| 0.4515|
|          | Best       | 0.4630| 0.4582| 0.4569| 0.4562| 0.4562|
|          | Mean       | 0.4510| 0.4492| 0.4556| 0.4562| 0.4562|
|          | SD         | 0.0087| 0.0074| 0.0016| 5.66E−17| 0.0012|
| Lung Cancer| Worst   | 0.4062| 0.4063| 0.5313| 0.4063| 0.5625|
|          | Best       | 0.5625| 0.5000| 0.5625| 0.6250| 0.5625|
|          | Mean       | 0.4641| 0.4281| 0.5578| 0.5188| 0.5625|
|          | SD         | 0.0433| 0.0250| 0.0114| 0.0644| 1.28E−16|

Figure 3: Benchmark tested images. (a) Church. (b) Train. (c) Roman. (d) Starfish. (e) Surfer II. (f) Cow. (g) Crocodile. (h) Elephant.
The number of \(i\)-th gray level pixels or frequencies is denoted by \(h(i)\), \(PR_i\) represents the probability of \(i\)-th gray level pixel in the image, \(PR_i = h(i)/N\), where \(N = \sum_{i=0}^{M} h(i)\). The optimal thresholds \(\{t_1, t_2, \ldots, t_M\}\) are determined by the following equations:

\[
\{t_1^*, t_2^*, \ldots, t_M^*\} = \arg \max \{\sigma_B^2(t_1, t_2, \ldots, t_M)\},
\]

\[
\sigma_B^2 = \sum_{k=1}^{M} w_k \left( u_k - u_k^* \right)^2,
\]

where \(w_k^2 = \sum_{i \in C_k} PR_i\), \(u_k = \sum_{i \in C_k} iPR_i/w_k\), for \(k = 1, 2, \ldots, M\).

The compared experiments are performed on the test images with a different number of thresholds, \(M = 3, M = 5, \) and \(M = 8\) [41], to evaluate the performance of the CCP in both low- and high-dimensional multilevel thresholding problems. And, the proposed CCP is compared with PSO, DE, EPSO, and APSO approaches as mention above. The purpose of image segmentation is to find a set of thresholds \(\{t_1, t_2, \ldots, t_M\}\) to maximize the values of Ostu’s function according to equation (17). Figures 4–7 give the segmented results on tested images with a different number of thresholds.

Figures 4–7 provide the segmented images on church, starfish, surfer II, and elephants obtained by the compared techniques. It can be observed that the segmented quality has been improved with the increasing number of thresholds. And, CCP has a better performance than others on these tested images. Respectively, the segmented images obtained by APSO and CCP achieve better consistency than those by PSO, DE, and EPSO when \(Th = 8\). The gray-level histogram is often regarded as a kind of distributions to determine the thresholds for the image segmentation [42]. And, the peak value of the histogram is one of the important factors that affect the segmentation accuracy. Therefore, the thresholds of the compared approaches on church and elephant images are shown in gray-level histogram as follows.

Figures 8 and 9 show the segmented results of church and elephant at 8 thresholds level obtained by CCP and compared approaches. The optimal thresholds values of CCP with \([40, 79, 114, 130, 158, 191, 213, 235]\) is similar to those of the PSO with \([40, 77, 114, 130, 157, 190, 213, 235]\) on church images and also similar to those of the elephant image. From these figures, it can be seen that the optimal thresholds values by DE and EPSO are very different from those by PSO, APSO, and CCP in most cases. Because the segmentation results depend on the information of classes according to the thresholds level, the optimal thresholds of EPSO have statistical difference with others. Furthermore, it is not difficult to find that the selected optimal thresholds heavily depend on the objective function that is chosen. Besides EPSO, other threshold selection approaches, including the CCP approach, can segment the test images more reasonably, as shown in Figures 4–7. Table 8 provides the mean and SD of Ostu’s function with 3, 5, and 8 thresholds achieved by all compared approaches in 50 run times.

The simple statistics of compared approaches on tested images are reported in Table 8. The best values of mean and SD for each image are highlighted, and it is not hard to see that the CCP system is able to find the best values. Because the PSO, DE, EPSO, and APSO are not specially techniques for image segmentation, some segmented approaches based on multilevel threshold, whale optimization algorithm (WOA) [43], gray wolf optimizer (GWO) [44], whale optimization algorithm based on thresholding heuristic (WOA-TH) [41], and gray wolf optimizer based on thresholding heuristic (GWO-TH) [41], are used to evaluate
the clustering effectiveness of CCP. The maximum number of iterations is set to 2200, 3000, and 3600 with the number of thresholds being 3, 5, and 8. The mean and SD of Ostu’s function are obtained by the 100 run times to avoid the effects of random factors.

Table 9 provides the values of mean and SD obtained by different segmented approaches on tested images. Obviously, traditional approaches are easily trapped into local optima, and WOA and GWO have worse segmented performance than that of the others. And, the thresholding heuristic has finetuned the best thresholds to enhance the global search ability of WOA and GWO. It also can be observed that the CCP has better and stable performance than compared segmented approaches on the tested images.

Figure 5: Segmented images of starfish at different thresholds calculated by all approaches: (a) PSO (Th = 3), (b) DE (Th = 3), (c) EPSO (Th = 3), (d) APSO (Th = 3), (e) CCP (Th = 3), (f) PSO (Th = 5), (g) DE (Th = 5), (h) EPSO (Th = 5), (i) APSO (Th = 5), (j) CCP (Th = 5), (k) PSO (Th = 8), (l) DE (Th = 8), (m) EPSO (Th = 8), (n) APSO (Th = 8), (o) CCP (Th = 8).

Figure 6: Segmented images of Surfer II at different thresholds calculated by all approaches: (a) PSO (Th = 3), (b) DE (Th = 3), (c) EPSO (Th = 3), (d) APSO (Th = 3), (e) CCP (Th = 3), (f) PSO (Th = 5), (g) DE (Th = 5), (h) EPSO (Th = 5), (i) APSO (Th = 5), (j) CCP (Th = 5), (k) PSO (Th = 8), (l) DE (Th = 8), (m) EPSO (Th = 8), (n) APSO (Th = 8), (o) CCP (Th = 8).
Figure 7: Segmented images of elephant at different thresholds calculated by all approaches: (a) PSO (Th = 3), (b) DE (Th = 3), (c) EPSO (Th = 3), (d) APSO (Th = 3), (e) CCP (Th = 3), (f) PSO (Th = 5), (g) DE (Th = 5), (h) EPSO (Th = 5), (i) APSO (Th = 5), (j) CCP (Th = 5), (k) PSO (Th = 8), (l) DE (Th = 8), (m) EPSO (Th = 8), (n) APSO (Th = 8), (o) CCP (Th = 8).

Figure 8: Continued.
Figure 8: Segmented images of church at 8 thresholds level calculated by all approaches and their respective gray histogram. (a) Church, (b) PSO, (c) DE, (d) EPSO, (e) APSO, and (f) CCP.

Figure 9: Continued.
Figure 9: Segmented images of elephant at 8 thresholds level calculated by all approaches and their respective gray histogram. (a) Elephant, (b) PSO, (c) DE, (d) EPSO, (e) APSO, and (f) CCP.

Table 8: The values of Ostu’s function with a different number of thresholds calculated by all approaches.

| Images     | Th | P   | Segmentation approaches |
|------------|----|-----|--------------------------|
|            |    |     | PSO         | DE         | EPSO        | APSO        | CCP         |
| Church     | 3  | Mean| 3276.8367 | 3276.837  | 3276.7685  | 3276.8367  | 3276.8367  |
|            |    | SD  | 1.87E − 12 | 1.87E − 12| 0.155831  | 1.87E − 12 | 0.00E + 00 |
|            | 5  | Mean| 3381.9107 | 3381.4923 | 3380.4929  | 3381.9236  | 3381.9246  |
|            |    | SD  | 0.0171    | 0.2245    | 1.1240    | 0.0046     | 9.25E − 13 |
|            | 8  | Mean| 3422.5621 | 3420.9578 | 3416.2082  | 3420.7834  | 3423.2006  |
|            |    | SD  | 2.6579    | 0.9181    | 3.5574    | 2.86725    | 1.2323     |
| Train      | 3  | Mean| 2606.5951 | 2606.5951 | 2606.5623  | 2606.5951  | 2606.5951  |
|            |    | SD  | 4.67E − 13| 4.67E − 13| 0.0500    | 4.67E − 13 | 4.67E − 13 |
|            | 5  | Mean| 2736.2257 | 2735.3181 | 2735.2676  | 2735.6328  | 2736.2331  |
|            |    | SD  | 0.0135    | 0.5241    | 0.6743    | 2.6787     | 9.33E − 13 |
|            | 8  | Mean| 2790.3435 | 2788.0338 | 2786.9829  | 2790.4068  | 2790.4461  |
|            |    | SD  | 0.0913    | 0.5724    | 2.5511    | 0.0897     | 0.0389     |
| Roman      | 3  | Mean| 2142.3440 | 2142.3440 | 2142.2752  | 2142.3440  | 2142.3440  |
|            |    | SD  | 9.33E − 13| 9.33E − 13| 0.1073    | 9.33E − 13 | 1.39E − 13 |
|            | 5  | Mean| 2221.3026 | 2220.8465 | 2220.5998  | 2221.3051  | 2221.3056  |
|            |    | SD  | 0.0047    | 0.2476    | 0.4636    | 0.0012     | 1.67E − 13 |
|            | 8  | Mean| 2251.7218 | 2250.5766 | 2249.7117  | 2251.7258  | 2251.7325  |
|            |    | SD  | 0.0592    | 0.4211    | 1.2147    | 0.0673     | 0.0573     |
| Starfish   | 3  | Mean| 2784.2272 | 2784.2272 | 2784.0795  | 2784.2272  | 2784.2272  |
|            |    | SD  | 1.4E − 12 | 1.4E − 12 | 0.1267    | 1.4E − 12  | 9.25E − 13 |
|            | 5  | Mean| 2916.2557 | 2915.6466 | 2915.5862  | 2916.2726  | 2916.2730  |
|            |    | SD  | 0.0240    | 0.3693    | 0.6716    | 0.0010     | 0.0009     |
|            | 8  | Mean| 2973.9976 | 2971.8186 | 2971.2207  | 2974.0391  | 2974.0829  |
|            |    | SD  | 0.0404    | 0.7603    | 1.8786    | 0.0942     | 0.0151     |
| Surfer II  | 3  | Mean| 7953.4181 | 7953.4181 | 7953.3489  | 7953.4181  | 7953.4181  |
|            |    | SD  | 2.80E − 12| 2.80E − 12| 0.0718    | 2.78E − 12 | 2.80E − 12 |
|            | 5  | Mean| 8025.5880 | 8025.1056 | 8024.6160  | 8025.6045  | 8025.6055  |
|            |    | SD  | 0.0147    | 0.2639    | 1.2232    | 0.0031     | 0.0019     |
|            | 8  | Mean| 8055.9764 | 8054.3387 | 8052.1159  | 8056.0443  | 8056.0530  |
|            |    | SD  | 0.0505    | 0.5497    | 2.2741    | 0.0361     | 0.0189     |
| Images     | Th | P   | Segmentation approaches |              |              |              |
|------------|----|-----|-------------------------|--------------|--------------|--------------|
|            |    |     | PSO                     | DE           | EPSO         | APSO         |
|            |    |     |                          |              |              |              |
| Cow        | 3  | Mean| 3858.2202               | 3858.2202    | 3858.1872    | 3858.2200    | 3858.2202    |
|            |    | SD  | 0.00E + 00              | 0.00E + 00   | 0.0886       | 0.0009       | 0.00E + 00   |
|            | 5  | Mean| 3965.2806               | 3964.3722    | 3964.0770    | 3965.2925    | 3965.2932    |
|            |    | SD  | 0.0196                  | 0.4531       | 1.2480       | 0.0032       | 4.67E - 13   |
|            | 8  | Mean| 4028.8447               | 4026.0490    | 4024.4213    | 4028.9299    | 4028.9450    |
|            |    | SD  | 0.0758                  | 0.8635       | 2.6447       | 0.0548       | 0.0107       |
| Crocodile  | 3  | Mean| 3155.4359               | 3155.4343    | 3155.3468    | 3155.4359    | 3155.4359    |
|            |    | SD  | 9.33E - 13              | 0.0069       | 0.0969       | 9.33E - 13   | 9.25E - 13   |
|            | 5  | Mean| 3291.4296               | 3290.6065    | 3290.4743    | 3291.4263    | 3291.4407    |
|            |    | SD  | 0.0163                  | 0.4943       | 1.0594       | 0.0032       | 0.00E + 00   |
|            | 8  | Mean| 3348.6314               | 3346.4809    | 3345.8010    | 3348.6425    | 3348.7305    |
|            |    | SD  | 0.0719                  | 0.6380       | 2.0889       | 0.1033       | 0.0168       |
| Elephants  | 3  | Mean| 1626.7205               | 1626.7205    | 1626.6710    | 1626.7198    | 1626.7205    |
|            |    | SD  | 0.00E + 00              | 0.00E + 00   | 0.1220       | 0.0035       | 0.00E + 00   |
|            | 5  | Mean| 1695.8663               | 1695.5298    | 1695.2335    | 1695.8663    | 1695.8676    |
|            |    | SD  | 0.0039                  | 0.1415       | 0.8755       | 0.0023       | 0.0005       |
|            | 8  | Mean| 1728.1706               | 1726.9276    | 1726.1534    | 1728.0408    | 1728.2503    |
|            |    | SD  | 0.0592                  | 0.3931       | 1.5228       | 0.6764       | 0.0232       |

| Church     | 3  | Mean| 3271.4427               | 3271.4383    | 3271.4427    | 3271.4445    | 3276.8367    |
| Train      | 3  | Mean| 2606.5951               | 2611.4987    | 2611.5081    | 2611.5081    | 2606.5951    |
| Roman      | 3  | Mean| 2138.7893               | 2138.7837    | 2138.8057    | 2138.7992    | 2142.3440    |
| Starfish   | 3  | Mean| 2779.9252               | 2779.9167    | 2779.9252    | 2779.9214    | 2784.2272    |
| Surfer II  | 3  | Mean| 7953.4167               | 7953.4063    | 7953.4167    | 7953.4135    | 7953.4181    |
|            |    | SD  | 5.07E - 03              | 1.89E - 02   | 4.82E - 03   | 7.80E - 03   | 6.48E - 12   |
|            | 5  | Mean| 8025.0703               | 8025.4959    | 8025.6049    | 8025.6025    | 8025.6055    |
|            |    | SD  | 7.73E - 03              | 5.56E - 02   | 7.11E - 03   | 3.20E - 03   | 2.80E - 12   |
|            | 8  | Mean| 8055.4215               | 8054.4135    | 8056.0617    | 8055.8463    | 8056.0646    |
|            |    | SD  | 8.27E - 03              | 7.36E - 01   | 2.05E - 03   | 1.25E - 01   | 2.01E - 03   |
5. Conclusions

A complex chained P system (CCP) is proposed for solving clustering problems, which combines complex chained P systems and evolution mechanisms, including PSO and DE mechanisms. Two kinds of evolution rules for objects in different chained membranes are introduced to enhance the global search ability of PSO. One of evolution rules contain two subevolution rules, which are based on traditional and modified PSO techniques. The partitioning information as environmental factor is introduced to improve the clustering performance of PSO. Another is based on the DE mechanism to evolve the global chained objects in the chained membrane $m+1$ to enhance the global search ability. In addition, two kinds of communication rules in the chained P systems are defined to enhance the cooperation between chained membranes and avoid prematurity. In order to verify the validity and the performance of CCP, this proposed system is evaluated on eight benchmark clustering problems from the UCI machine learning repository as compared with four developed clustering approaches. Furthermore, eight tested images which from the Berkeley segmentation image databases BSDS300 are used to further evaluate the performance of CCP compared to four existing segmentation techniques. These experimental results verify the validity and performance of this proposed CCP.

P systems, as parallel computing models, are highly effective and efficient in solving optimization problems with linear or polynomial complexity. These parallel computing models based on evolution mechanisms provide new ways for solving clustering problems. The extended clustering chained P system uses the chained P system as the computation structure, and the communication rules between chained membranes are single directional. Although these single directional communication rules are simple and easy to implement, bidirectional communication rules may be introduced in future studies to further accelerate the convergence and improve the diversity of populations. Some more complicated communication structures between different membranes may be used in future studies to improve the performance of the approach. Furthermore, the experiments only used small datasets from the artificial datasets and the UCI Machine Learning Repository, and the proposed approach may have some limitations on high dimensional and large datasets. Future studies may test the effectiveness of CCP using large datasets. Balancing the local and global search abilities is also a hard problem to resolve in the future studies. Future studies may also focus on extended P systems based on tissue-like P systems and other bioinspired computing models. More works are needed to apply these extended membrane systems to solve automatic and multiojbective clustering problems.

Data Availability

The two artificial datasets that were manually generated and often used in the existing literature are from the artificial datasets, available at https://www.isical.ac.in/content/databases (accessed June 2018). The eight real-life datasets are often used in the existing literature from the UCI Machine Learning Repository, available at http://archive.ics.uci.edu/ml/datasets.html (accessed June 2018). The eight tested images are from the Berkeley computer vision group, Berkeley segmentation dataset, and benchmark (BSDS300), available at https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench (accessed October 2018).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Table 9: Continued.

| Images       | Th | P            | WOA          | GWO          | WOA-TH        | GWO-TH        | CCP          |
|--------------|----|--------------|--------------|--------------|--------------|--------------|--------------|
| Cow          | 3  | Mean         | 3858.2202    | 3858.2188    | 3858.2202    | 3858.2202    | 3858.2202    |
|              |    | SD           | 5.94E-12     | 1.03E-02     | 5.94E-12     | 5.94E-12     | 1.85E-12     |
|              | 5  | Mean         | 3965.2795    | 3965.1739    | 3965.2932    | 3964.6096    | 3965.2932    |
|              |    | SD           | 2.01E-02     | 9.72E-02     | 6.40E-12     | 2.93E+00     | 1.39E-12     |
|              | 8  | Mean         | 4028.9410    | 4027.7892    | 4028.8474    | 4027.7210    | 4028.9605    |
|              |    | SD           | 1.83E-02     | 9.47E-01     | 6.11E-01     | 1.69E+00     | 1.50E-02     |
| Crocodile    | 3  | Mean         | 3155.4359    | 3155.4337    | 3155.4359    | 3155.4359    | 3155.4359    |
|              |    | SD           | 9.14E-13     | 1.39E-02     | 9.14E-13     | 9.14E-13     | 9.33E-13     |
|              | 5  | Mean         | 3291.4340    | 3291.2945    | 3291.4366    | 3291.4287    | 3291.4407    |
|              |    | SD           | 1.02E-02     | 1.22E-01     | 5.38E-03     | 1.35E-02     | 4.67E-13     |
|              | 8  | Mean         | 3348.0384    | 3347.4528    | 3348.7453    | 3348.6504    | 3348.7454    |
|              |    | SD           | 2.79E+00     | 1.75E+00     | 1.94E-03     | 7.60E-02     | 3.27E-03     |
| Elephants    | 3  | Mean         | 1626.7183    | 1626.7192    | 1626.7205    | 1626.7196    | 1626.7205    |
|              |    | SD           | 3.33E-03     | 9.10E-03     | 1.37E-12     | 9.25E-03     | 9.25E-13     |
|              | 5  | Mean         | 1695.0482    | 1695.3208    | 1695.8664    | 1695.8231    | 1695.8676    |
|              |    | SD           | 3.99E+00     | 2.84E+00     | 1.05E-03     | 8.42E-02     | 7.12E-02     |
|              | 8  | Mean         | 1727.1725    | 1726.5379    | 1728.2596    | 1728.1452    | 1728.2676    |
|              |    | SD           | 2.86E+00     | 1.34E+00     | 2.55E-02     | 1.35E-01     | 1.20E-02     |
Acknowledgments

This research project was partially supported by the National Natural Science Foundation of China (61876101, 61802234, and 61806114), the National Natural Science Foundation of Shandong Province, China (ZR2019QF007), the Ministry of Education Humanities and Social Science Research Youth Foundation, China (19YJCZH244), the Special Science Fund Project of Shandong Province, China (16BGLJ06 and 11CGLJ22), the Special Postdoctoral Project of China (2019T120607), and Postdoctoral Project of China (2017M612339 and 2018M642695).

References

[1] G. H. Păun, “Membrane computing: an introduction,” Theoretical Computer Science, vol. 287, no. 1, pp. 73–100, 2002.
[2] X. Song, J. Wang, H. Peng et al., “Spiking neural P systems with multiple channels and anti-spikes,” BioSystems, vol. 170, pp. 13–19, 2018.
[3] A. Leporati, L. Manzoni, G. Mauri et al., “The counting power of P systems with antimatter,” Theoretical Computer Science, vol. 701, no. 21, pp. 161–173, 2017.
[4] A. Mahmood, A. Maroosi, and R. C. Muniyandi, “Membrane computing to enhance time efficiency of minimum dominating set,” Mathematics in Computer Science, vol. 10, no. 2, pp. 249–261, 2016.
[5] A. Leporati, L. Manzoni, and G. Mauri, “Characterising the complexity of tissue P systems with fission rules,” Journal of Computer and System Sciences, vol. 90, pp. 115–128, 2017.
[6] L. Q. Pan, G. H. Păun, and B. Son, “Flat maximal parallelism in P systems with promoters,” Theoretical Computer Science, vol. 623, pp. 83–91, 2016.
[7] A. Alhazov, R. Freund, and S. Ivanov, “Extended spiking neural P systems with white hole rules and their red-green variants,” Natural Computing, vol. 17, no. 2, pp. 297–310, 2017.
[8] T. Pan, X. Shi, and Z. Zhang, “A small universal spiking neural P system with communication on request,” Neurocomputing, vol. 275, pp. 1622–1628, 2018.
[9] L. Wang, X. Liu, and M. Sun, “An Extended clustering membrane system based on particle swarm optimization and cell-like P system with active membranes,” Mathematical Problems in Engineering, vol. 2020, Article ID 5097589, 18 pages, 2020.
[10] X. Y. Liu and J. Xue, “A Cluster splitting technique by hopfield networks and P systems on simplices,” Neural Processing Letters, vol. 46, no. 1, pp. 171–194, 2017.
[11] X. Liu, Y. Zhao, and M. Sun, “An improved Apriori algorithm based on an evaluation-communication tissue-like P system with promoters and inhibitors,” Discrete Dynamics in Nature and Society, vol. 2017, no. 4, pp. 1–11, 2017.
[12] H. Peng, J. Wang, and P. Shi, “An extended membrane system with active membranes to solve automatic fuzzy clustering problems,” International Journal of Neural Systems, vol. 26, no. 3, Article ID 1650004, 2016.
[13] H. Peng, P. Shi, and J. Wang, “Multiobjective fuzzy clustering approach based on tissue-like membrane systems,” Knowledge-Based Systems, vol. 125, pp. 74–82, 2017.
[14] Y. H. Wang, X. Y. Liu, and L. S. Xiang, “GA-based membrane evolutionary algorithm for ensemble clustering,” Computational Intelligence and Neuroscience, vol. 2017, Article ID 4367342, 11 pages, 2017.
[15] M. Maitra and C. Amitava, “A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding,” Expert Systems with Applications, vol. 34, no. 2, pp. 1341–1350, 2008.
[16] A. Mur Angel, R. Dormido, and N. Duro, “Determination of the optimal number of clusters using a spectral clustering optimization,” Expert Systems with Applications, vol. 65, pp. 304–314, C, 2016.
[17] S. Shirly and K. Ramesh, “Review on 2D and 3D MRI image segmentation techniques,” Current Medical Imaging Formerly Current Medical Imaging Reviews, vol. 15, no. 2, pp. 150–160, 2019.
[18] M. Zhang, L. Zhang, and H. D. Cheng, “A neuroptic approach to image segmentation based on watershed method,” Signal Processing, vol. 90, no. 5, pp. 1510–1517, 2010.
[19] T. P. Xuan, P. Siarry, and H. Oulhadj, “Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation,” Applied Soft Computing, vol. 65, pp. 230–242, 2018.
[20] R. Rahmat and D. B. Harris-Birtill, “Comparison of level set models in image segmentation,” IET Image Processing, vol. 12, no. 12, pp. 2212–2221, 2018.
[21] J. L. Pei, L. Zhao, and X. J. Dong, “Effective algorithm for determining the number of clusters and its application in image segmentation,” Cluster Computing, vol. 20, no. 4, pp. 2845–2854, 2017.
[22] J. F. Li, W. Y. Tang, and J. Wang, “Multilevel thresholding selection based on variational mode decomposition for image segmentation,” Signal Processing, vol. 147, pp. 80–91, 2018.
[23] X. L. Zao, J.-H. Hwang, and Z. J. Fang, “Gradient-based adaptive particle swarm optimizer with improved extremal optimization,” Applied Intelligence, vol. 48, no. 12, pp. 4646–4659, 2018.
[24] B. Wang, L. L. Chen, and J. Cheng, “New result on maximum entropy threshold image segmentation based on P System,” Optik, vol. 163, pp. 81–85, 2018.
[25] T. Y. Tan, L. Zhang, and C. P. Lim, “Evolving ensemble models for image segmentation using enhanced particle swarm optimization,” IEEE Access, vol. 7, pp. 34004–34019, 2019.
[26] R. Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in Proceedings of the 2001 Congress on Evolutionary Computation, pp. 81–86, Seoul, South Korea, South, May 2001.
[27] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.
[28] T. F. Araújo and U. Waded, “Performance assessment of PSO, DE and hybrid PSO-DE algorithms when applied to the dispatch of generation and demand,” International Journal of Electrical Power & Energy Systems, vol. 47, pp. 205–217, 2013.
[29] Y. Z. Zhao, X. Y. Liu, and W. X. Sun, “The chained P system with application in graph clustering,” in Proceedings of the Sixth Asian Conference on Membrane Computing, pp. 46–57, Chengdu, China, September 2017.
[30] D. E. Goldberg, “Genetic algorithm in search, optimization and machine learning,” 1989.
[31] A. H. Cao, “A passive location algorithm based on differential evolution and genetic algorithm using the Doppler frequency,” Signal Processing, vol. 25, no. 10, pp. 1644–1648, 2009.
[32] S. H. Liao, C. C. Chiu, and M. H. Ho, “Comparison of dynamic differential evolution and genetic algorithm for
MIMO-WLAN transmitter antenna location in indoor environment,” *Wireless Personal Communications*, vol. 71, no. 4, pp. 2677–2691, 2009.

[33] Datasets, http://www.isical.ac.in/content/database.

[34] H. C. Blake, “UCI repository of machine learning databases,” 1998.

[35] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Western Australia, November 1995.

[36] R. Storn and P. Kenneth, “Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[37] W. Song, W. Ma, and Y. G. Qiao, “Particle swarm optimization algorithm with environmental factors for clustering analysis,” *Soft Computing*, vol. 21, no. 2, pp. 283–293, 2017.

[38] H.-G. Han, W. Lu, and Y. Hou, “An adaptive-PSO-based self-organizing RBF neural network,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 104–117, 2018.

[39] The berkeley segmentation dataset and benchmark,” https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/.

[40] N. Qtsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[41] B. V. Kumar and K. V. Arya, “A new heuristic for multilevel thresholding of images,” *Expert Systems with Applications*, vol. 117, pp. 176–203, 2019.

[42] D. P. Panda and A. Rosenfeld, “Image segmentation by pixel classification in (gray level, edge value) space,” *IEEE Transactions on Computers*, vol. C-27, no. 9, pp. 875–879, 1978.

[43] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[44] S. Mirjalili, M. Seyed, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, no. 3, pp. 46–61, 2014.