

Research Article

An Extended Clustering Membrane System Based on Particle Swarm Optimization and Cell-Like P System with Active Membranes

Lin Wang ^{1,2}, Xiyu Liu ^{1,2}, Minghe Sun ³, and Jianhua Qu ^{1,2}

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

²College of Business, Shandong Normal University, Jinan, China

³College of Business, The University of Texas at San Antonio, San Antonio, TX, USA

Correspondence should be addressed to Xiyu Liu; sdxyliu@163.com

Received 1 October 2019; Revised 23 December 2019; Accepted 11 January 2020; Published 31 January 2020

Academic Editor: Pietro Bia

Copyright © 2020 Lin Wang 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.

An extended clustering membrane system using a cell-like P system with active membranes based on particle swarm optimization (PSO), named PSO-CP, is designed, developed, implemented, and tested. The purpose of PSO-CP is to solve clustering problems. In PSO-CP, evolution rules based on the standard PSO mechanism are used to evolve the objects and communication rules are adopted to accelerate convergence and avoid prematurity. Subsystems of membranes are generated and dissolved by the membrane creation and dissolution rules, and a modified PSO mechanism is developed to help the objects escape from local optima. Under the control of the evolution-communication mechanism, the extended membrane system can effectively search for the optimal partitioning and improve the clustering performance with the help of the distributed parallel computing model. This extended clustering membrane system is compared with five existing PSO clustering approaches using ten benchmark clustering problems, and the computational results demonstrate the effectiveness of PSO-CP.

1. Introduction

The scopes and scales of datasets are growing exponentially with the advent of new sources of data generation. This growing tendency produces a serious challenge for discovering knowledge from data. Data clustering is one of the important techniques used in data mining [1]. It aims to put similar data points into the same group or cluster using the characteristics of the data without any prior knowledge about the groups or clusters. Therefore, the implicit patterns or knowledge can be extracted through data clustering [2]. Traditional data clustering approaches can be categorized into partition clustering, hierarchical clustering, density clustering, and grid clustering [3]. These clustering methods have low time complexity and are easy to implement, but also produce highly skewed dendrograms that may not reflect the true structures of the datasets [4, 5]. Therefore, some evolutionary approaches have been introduced to solve

clustering problems in recent years [6], such as genetic algorithms [7], particle swarm optimization (PSO) [8], differential evolution (DE) [9], artificial bee colony (ABO) [10], and ant colony optimization (ACO) [11], among others. PSO is one of the global optimization techniques based on the intelligence strategy of the population, and many works have been done to use PSO to solve clustering problems.

Netjinda et al. [12] presented a PSO approach, named starling PSO (SPSO), which is inspired by the collective response behavior of starling birds. The collective information of the neighbors is used to replace the local information in history, and subpopulations are generated when the premature phenomenon appears. Song et al. [13] proposed an improved PSO procedure based on the features of the clustered data. An environment factor is added to the velocity adjustment in PSO to improve the global searching ability, which is represented by the cluster centers of the partitioning results. Liu et al. [14] developed a modified

coevolutionary multiswarm optimizer based on a new velocity updating and similarity detection mechanism. Lassad et al. [15] designed a PSO procedure with new adaptive inertia weight and time acceleration coefficients for solving fuzzy clustering problems. Asgarali and Abdolreza [16] introduced a K -harmonic means clustering approach, which integrates the improved cuckoo search and PSO. Pereira de Gusmão and de Carvalho [17] proposed two hybrid clustering methods for multiview relational data, taking advantages of the global convergence ability of PSO and the local exploitation of hard clustering algorithms in the update of the position vectors in PSO. Manju and Kumar [18] developed a new sustainable clustering method based on PSO with a mutation operator for clustering of the data generated from different networks. Huang et al. [19] designed a memetic clustering approach based on PSO and the gravitational search algorithm using the hybrid operation and a diversity enhancement as the two main mechanisms. Zhang et al. [20] presented a new clustering approach based on PSO with a leader updating mechanism and ring topology for multimodal multiobjective optimization.

Although PSO has shown a great potential in solving clustering problems, it still has some limitations, such as being easily falling into local optima and exhibiting the premature phenomena. Furthermore, the computational complexity of the PSO clustering approaches may increase quickly as the number of data points in the dataset increases [21]. Therefore, more studies are needed to improve the performance of PSO for clustering.

Membrane computing, also known as membrane systems or P systems, is a novel approach of bio-inspired computing initiated by Păun [22]. It seeks to discover novel biological computing models from the structure of biological cells as well as the cooperation of cells in tissues and organs. Parallel computation in membrane systems can avoid the increase in time consumption with the increase in the number of data points. Therefore, membrane systems are suitable for solving clustering problems [23]. Research shows that some models of P systems present the same computing power as Turing machines and are more efficient to some extent [24]. Spiking neural P systems are a kind of neural-like P systems in membrane computing. It provides a class of parallel computing models [25]. Many variants of spiking neural P systems have been proposed [26, 27] and have been applied to various real-world problems [28, 29].

Xue and Liu [30] developed a new communication P system for solving clustering problems. Liu and Xue [31] proposed a new cluster splitting technique based on Hopfield neural networks and P systems. Liu et al. [32] presented an improved *Apriori* algorithm, named ECTPPT-*Apriori*, based on evolution-communication tissue-like P systems with promoters and inhibitors. Peng et al. [33] designed a tissue-like membrane system with a fully connected structure using an inherent mechanism to deal with automatic clustering problems. Peng et al. [34] developed an extended membrane system with active membranes, in which a modified differential evolution mechanism is used to find the optimal cluster centers in clustering problems. Peng et al.

[35] introduced a multiobjective clustering framework using a tissue-like membrane system for fuzzy clustering problems. Wang et al. [36] proposed a new cell-like P clustering system using a modified genetic algorithm to evolve the objects and using communication rules in the cell-like P system to enhance the diversity of the populations.

The traditional evolution mechanism is easily trapped into local optima, called the premature phenomenon, which is a main limitation of PSO for solving optimization problems. Many previous studies paid close attention to improving the global searching ability and avoiding prematurity. Membrane systems are distributed parallel computing models and can effectively avoid the prematurity and improve the global searching ability of PSO. Over the past years, a variety of membrane systems integrated with PSO have been proposed and proved powerful and efficient in solving optimization problems. Xiao et al. [37] proposed a hybrid membrane evolutionary algorithm, which combines a one-level membrane structure with a PSO local search algorithm. Xiao et al. [38] developed an improved dynamic membrane evolutionary algorithm based on PSO and DE to solve constrained engineering design problems. Singh and Deep [39] designed a new multiple-PSO based membrane algorithm with seven different membranes for solving real-life problems. Elkhani et al. [40] proposed a kernel P system and introduced multiobjective binary PSO to feature selection and classification methods with time efficiency on GPU. Furthermore, the inherent mechanism based on communication rules between different cells or membranes can accelerate convergence of PSO. Therefore, membrane systems are used to enhance the clustering performance of PSO in this study. Each cell or membrane in a membrane system, as an independent computing unit, can be regarded as a subpopulation of particles in PSO, and the cooperation of subpopulations can be viewed as the communication between membranes [41].

This work focuses on the development of a membrane computing model, as an extended membrane system, to solve clustering problems and overcome the limitations mentioned above. A new clustering method based on membrane systems and the PSO mechanism is proposed. This membrane system with active membranes has a dynamic membrane structure during evolution and computation. The velocity updating mechanism of PSO is used as the basic evolution rules for the objects in the elementary membranes. Another evolution rule based on the fitness Euclidean-distance ratio (FER) [42] method is introduced for the objects to escape the local optima in the membranes of the subsystems. The communication mechanism in membranes is adopted to transport the best objects in order to accelerate the convergence of the P system. This system is evaluated using 10 benchmark clustering problems to verify the validity and performance of the extended clustering membrane system.

The rest of this paper is organized as follows. The clustering problems are described in Section 2. The framework of cell-like P systems with active membranes is given in Section 3. These concepts are related to the development of the proposed extended clustering membrane

system. Section 4 describes the details of the extended clustering membrane system based on cell-like P systems and the PSO mechanism. Experimental results on benchmark clustering problems are reported in Section 5. Section 6 provides conclusions and outlines future research directions.

2. Data Clustering

In this section, the basic concepts of data clustering problems are described in detail. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be a dataset containing N unlabelled data points. Data point i , for $i = 1, 2, \dots, N$, is represented by $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ with d representing the dimension of the data. The purpose of a clustering problem is to find a partition of the dataset with similar data points in the same cluster. The partition result is represented by $C = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$, where K is the number of clusters and \mathbf{c}_k is cluster k , for $k = 1, 2, \dots, K$. The vector of the cluster centers is represented by $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K\}$ with \mathbf{z}_k representing the cluster center of \mathbf{c}_k [43].

A partition must satisfy some conditions, such as the data points in the same cluster should be similar as much as possible and the data points in different clusters should be different as much as possible. Usually, a clustering technique may search in the solution space to find the optimal cluster centers based on some clustering measures. A commonly used clustering measure, called the fitness function, is defined as follows:

$$f(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K) = \sum_{i=1}^N \sum_{j=1}^K \omega_{ij} \|\mathbf{x}_i - \mathbf{z}_j\|, \quad (1)$$

where ω_{ij} is the associated weight for data point \mathbf{x}_i to belong to the cluster j . If the data point \mathbf{x}_i is allocated to cluster j , $\omega_{ij} = 1$; otherwise, $\omega_{ij} = 0$. A clustering process is to separate the data points into the corresponding clusters that can be viewed as an optimization problem, and the purpose of the optimization problem is to find a partition or a set of cluster centers to minimize the fitness function (1), i.e.,

$$\min_{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K} f = \min f(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K) = \min \sum_{i=1}^N \sum_{j=1}^K \omega_{ij} \|\mathbf{x}_i - \mathbf{z}_j\|. \quad (2)$$

In addition, the value of the fitness function f is used to evaluate the performance of clustering techniques and to compare the quality of objects or potential solutions. When two objects are compared, the one with a smaller value of the fitness function is better than the other.

3. Cell-Like P Systems

3.1. The Basic Cell-Like P Systems. Cell-like P systems are a class of membrane systems, which abstract computing models from cell structures and functions or from the group collaboration of cells. Research shows that the computation ability of simple cell-like P systems is equal to that of Turing machines [44]. The usual cell-like P systems have a tree membrane structure, that is, a simple graph. Each membrane

contains a set of objects or symbols that can be evolved and communicated by evolution and communication rules. A basic cell-like P system can be expressed as the following tuple:

$$\prod = (O, H, \mu, w_1, \dots, w_q, R, R', i_0), \quad (3)$$

where $q \geq 1$ is the degree of the system; O is a finite set of alphabets, whose symbols are called objects, i.e., u and v represent different objects in the alphabets, where $u, v \in O$; H is a finite set of labels for the membranes; μ is the membrane structure consisting of q membranes and its regions are labelled by the elements of H ; w_1, \dots, w_q are the multisets of objects placed in the regions of the membranes, with $w_i \in O$, for $1 \leq i \leq q$; R represents multiple but finite sets of evolution rules associated with the membranes; R' represents multiple but finite sets of communication rules between different membranes; and i_0 is the output region or output membrane in the P system [45].

A cell-like P system is a hierarchy of q membranes or cells where each membrane or cell may contain one or more other membranes or cells. A membrane or cell contains many objects in the system, and an object u represents a potential solution in the search space, where $u \in O$. A membrane is called an *elementary membrane* if it does not contain any other membranes. An elementary membrane has no children membranes in the system. A membrane is called a *nonelementary membrane* if it contains other membranes. A membrane is called a *skin membrane* if it is not contained in any other membranes. The skin membrane has no parent membrane in the system. The degree of the system is the number of elementary membranes in the system. A cell-like P system is mainly composed of three parts: membrane structure, objects, and rules. Figure 1(a) gives a graphical representation of a simple cell-like P system, in which membranes are labelled from 0 to 9. These membranes are arranged by a hierarchical structure which can be represented as a tree diagram as shown in Figure 1(b).

In Figure 1(a), membrane 0 is the skin membrane which is not contained in any other membranes and membrane 1 is the parent membrane of membranes 4 and 5 and is a nonelementary membrane. Membrane 4 is an elementary membrane which does not contain any other membranes. The tree structure in Figure 1(b) is an abstract from the cell-like P system in Figure 1(a). The nodes in the tree represent the membranes, the leaf nodes represent the elementary membranes, and the root node 0 represents the skin membrane 0. A node in a layer is the parent of the nodes following it in the next layer, and the nodes in the next layer are the children of the node in the layer above. The degree of this cell-like P system is 6. Specially, a membrane only communicates with its parent membrane and children membranes, if it has any, and there are not existing communication rules between the sibling membranes.

A cell-like P system usually has two types of rules: evolution rules R and communication rules R' . Evolution rules are of the form $R = \{u \rightarrow v\}$, for $u, v \in O$, which means that a copy of object u will be evolved to object v . Communication rules are of the form $R' = \{u \rightarrow (v, in_h)\}$,

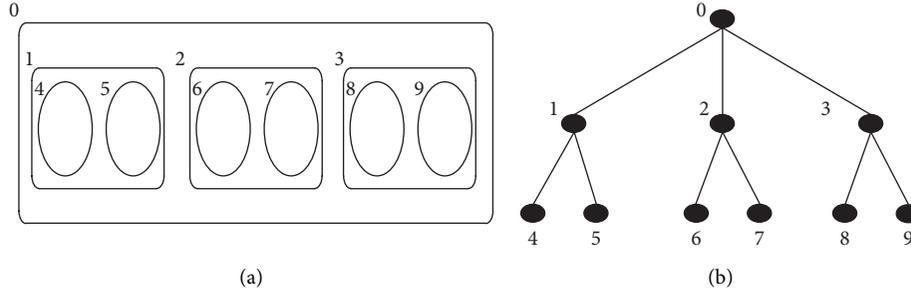


FIGURE 1: An example of cell-like P system.

for $h \in H$ and $u, v \in O$, which means that a copy of object u will be changed to object v and transported into membrane h . The object is modified in the communication process, and membrane h is the parent or child membrane of the membrane where originally u was.

3.2. An Extended Cell-Like P System with Active Membranes. The evolution and communication rules in the cell-like P systems only execute on objects, but not on membranes. The objects will be changed and moved based on the evolution-communication rules, but the membranes will not change during the evolution and computation. Therefore, an extended cell-like P system with active membranes is introduced to overcome this restriction. This extended P system contains not only evolution and communication rules for objects but also evolution rules for membranes.

There are two types of membrane evolution rules: creation rules and dissolution rules. Membrane creation rules are of the form $[u]_h \longrightarrow [v]_{h_1}, \dots, [v]_{h_{sn}}$, for $h, h_1, \dots, h_{sn} \in H$ and $u, v \in O$, which means that membrane h_1 to h_{sn} are created and a copy of object u in membrane h is evolved to v and transported to these newly created membranes, where sn is the number of newly created membranes. Membrane dissolution rules are of the form $[u]_h \longrightarrow \lambda$, for $u, \lambda \in O$, which means that the membrane will be dissolved, and object u in membrane h will disappear, where λ is a special symbol that represents no objects in the membrane. Therefore, the extended cell-like P system with active membranes has a dynamic membrane structure in the evolution and computation process [46].

4. An Extended Clustering Membrane System

An extended membrane system with active membranes, called the particle swarm optimization cell-like P system (PSO-CP), is introduced to solve clustering problems. This system has two main mechanisms: the evolution-

communication mechanism for objects and the evolution mechanism for membranes. More details about PSO-CP are given in the following.

4.1. Initialization

4.1.1. The Initial Membrane Structure. The membrane structure of PSO-CP is built dynamically through membrane creation and dissolution rules during the evolution and computation process. Therefore, the number of membranes will change. Specifically, PSO-CP starts with an initial membrane structure and then the membrane evolution mechanism will control the structure and the number of membranes. The initial membrane structure of a PSO-CP is graphically depicted in Figure 2.

In Figure 2, the PSO-CP with a three-layer nesting structure contains a skin membrane 0, a nonelementary membrane 1, and elementary membranes 2 to q . These membranes are labelled from 0 to q . The nonelementary membrane, also called the comparison membrane, is labelled 1, whose role is to find the best object in the elementary membranes during the current evolution process and output the best object to the outmost membrane. Membrane 1 is the parent of the elementary membranes 2 to q , and elementary membranes 2 to q are the children of nonelementary membrane 1. The outmost membrane, labelled 0, is the skin membrane, whose role is to store the best object in the system during the current evolution and computation process.

4.1.2. Object Representation. In the PSO-CP, an object is a set of clustering centers representing a feasible solution. Therefore, the objects represent the sets of clustering results. Each object u , $u \in O$, is designed as a composite $K \times d$ dimensional vector [47] of the following form:

$$u = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K\} = \{z_{11}, z_{12}, \dots, z_{1d}, \dots, z_{k1}, z_{k2}, \dots, z_{kd}, \dots, z_{K1}, z_{K2}, \dots, z_{Kd}\}, \quad (4)$$

where $\mathbf{z}_k = (z_{k1}, z_{k2}, \dots, z_{kd})$ corresponds to the cluster center of cluster k and d , as mentioned earlier, represents the dimension of the data points. Hence, an object u , with

$u = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K\}$, represents a set of cluster centers. As usual, each membrane has at least one object. To ensure the same computation complexity of each elementary

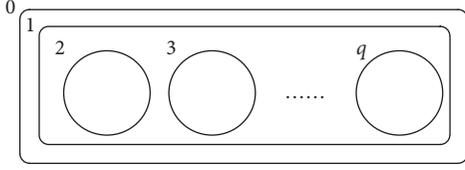


FIGURE 2: The initial structure of a PSO-CP.

membrane, all elementary membranes have the same number of objects, denoted by m . The number of objects in the whole P system is represented by M , i.e., $M = m \times q$.

4.1.3. Initial Objects. As mentioned above, each elementary membrane in the PSO-CP has the same number of objects. When the evolution and computation process starts, each elementary membrane also has the same number of initial objects. An object u represents a set of cluster centers. Dimension j of object u_p represented by u_{pj} is generated randomly between x_j^{\min} and x_j^{\max} , where x_j^{\min} and x_j^{\max} are the minimum and maximum values of dimension j of the search space. Therefore, $u_{pj} \in [x_j^{\min}, x_j^{\max}]$, for $p = 1, 2, \dots, m$ and $j = 1, 2, \dots, d$. The values of x_j^{\min} and x_j^{\max} are determined by using the following equation:

$$\begin{aligned} x_j^{\min} &= \min\{x_{ij} \mid i = 1, 2, \dots, N\}, \\ x_j^{\max} &= \max\{x_{ij} \mid i = 1, 2, \dots, N\}, \quad \text{for } j = 1, 2, \dots, d. \end{aligned} \quad (5)$$

After initialization, the fitness value of each object is calculated using (1). In membrane o , for $o = 2, 3, \dots, q$, the initial best with the lowest fitness value for u_p , denoted by $u_p^{\text{lbest}}(0)$ and called the local best, is determined for $p = 1, 2, \dots, m$. The initial best with the lowest fitness value in history among all objects in membrane o , denoted by $u_o^{\text{gbest}}(0)$ and called the global best, is also determined. The local best and the global best refer to the positions of the objects. The object that found the global best is the global best object.

4.2. The Evolution and Communication Rules for the Objects

4.2.1. Evolution Rules for the Objects. The PSO-CP has two types of evolution rules: the basic evolution and the local evolution rules.

(1) The Basic Evolution Rules. In this work, the standard PSO [48] is used to search for the optimal solutions in the elementary membranes 2 to q . The evolution of objects is achieved only within elementary membranes, and the basic evolution rules only execute on objects contained in elementary membranes 2 to q . Let $u_p(t)$ and $V_p(t)$ represent the position and velocity of u_p at time t in elementary membrane o , for $o = 2, 3, \dots, q$. The velocity of u_p at time $t+1$ is determined by using the following equation:

$$\begin{aligned} V_p(t+1) &= wV_p(t) + c_1r_1(u_p^{\text{lbest}}(t) - u_p(t)) \\ &\quad + c_2r_2(u_o^{\text{gbest}}(t) - u_p(t)), \end{aligned} \quad (6)$$

where w is the inertia weight, c_1 and c_2 represent the local and global learning factors, which control the influence of the local best and the global best object, t is the time or iteration counter, and r_1 and r_2 are two uniform random numbers [49]. The local best of u_p is denoted by $u_p^{\text{lbest}}(t)$ and the global best in the elementary membrane o is denoted by $u_o^{\text{gbest}}(t)$ at iteration t . For notational convenience, the global best object is also denoted by $u_o^{\text{gbest}}(t)$ at iteration t .

The position of u_p at time $t+1$ is determined by using the following equation:

$$u_p(t+1) = u_p(t) + V_p(t+1). \quad (7)$$

The local best of u_p at time $t+1$ is updated according to the following equation:

$$u_p^{\text{lbest}}(t+1) = \begin{cases} u_p(t+1), & f(u_p(t+1)) < f(u_p^{\text{lbest}}(t)), \\ u_p^{\text{lbest}}(t), & \text{otherwise.} \end{cases} \quad (8)$$

The global best in membrane o at iteration $t+1$ is updated according to the following equation:

$$u_o^{\text{gbest}}(t+1) = \begin{cases} u_p^{\text{lbest}}(t+1), & f(u_p^{\text{lbest}}(t+1)) < f(u_o^{\text{gbest}}(t)), \\ u_o^{\text{gbest}}(t), & \text{otherwise.} \end{cases} \quad (9)$$

The inertia weight is updated dynamically to enhance the global searching ability of the objects and to avoid premature. A linear increasing approach, given by (10), is used to update the inertia weight:

$$w(t) = w_{\min} + (w_{\max} - w_{\min}) * \left(\frac{t}{t_{\max}} \right), \quad (10)$$

where w_{\min} and w_{\max} represent the minimum and maximum of the inertia weight and t_{\max} is the maximum number of iterations.

(2) The Local Evolution Rules. A local search strategy based on the FER is adopted in local evolution rules for the objects. The approaches of updating the velocity and position of an object are modified. The local evolution of the objects is achieved only within a subsystem of membranes to help objects escape from local optima, and the local evolution rules only execute on objects contained in the membranes of the subsystem. The modified velocity is determined by using the following equation:

$$\begin{aligned} V_p(t+1) &= \chi(V_p(t) + r_3(u_p^{\text{lbest}}(t) - u_p(t)) \\ &\quad + r_4(u_p^{\text{nbest}}(t) - u_p(t))), \end{aligned} \quad (11)$$

and the modified position is determined by using the following equation:

$$u_p(t+1) = u_p(t) + V_p(t+1), \quad (12)$$

where χ is a constriction coefficient used to prevent an object from evolving too far away from the search space and is given by $\chi = 2/|2 - \varphi_{\max} - \sqrt{\varphi_{\max}^2 - 4\varphi_{\max}}|$ and r_3 and r_4 are two random numbers uniformly distributed between $[0, \varphi_{\max}/2]$. In the above, φ_{\max} is a positive constant with $\varphi_{\max} = 4.1$ [42]. In (11), $u_p^{\text{best}}(t)$ represents the best neighbor of u_p in the neighborhood, where the neighborhood is a subset of objects in the current evolution process. The objects in a membrane of the subsystem are listed in a decreasing order of the FER values. The value of the FER for a given u_p and any other $u_{p'}$, for $p' \neq p$ and $p' = 1, 2, \dots, m$, is determined by using the following equation:

$$\text{FER}(u_p(t), u_{p'}(t)) = \alpha \frac{f(u_p^{\text{best}}(t)) - f(u_{p'}^{\text{best}}(t))}{\|u_p^{\text{best}}(t) - u_{p'}^{\text{best}}(t)\|}, \quad (13)$$

where $\|u_p^{\text{best}}(t) - u_{p'}^{\text{best}}(t)\|$ is the Euclidean distance between $u_p^{\text{best}}(t)$ and $u_{p'}^{\text{best}}(t)$ at the current iteration t . Apparently, u_p and $u_{p'}$ only exist in the same membrane o . In (13), α represents a control parameter given by $\alpha = \|D\| / (f(u_o^{\text{worst}}(t)) - f(u_o^{\text{best}}(t)))$, where $u_o^{\text{worst}}(t)$ represents the worst objects with the highest value of the fitness function in membrane o , $u_o^{\text{best}}(t)$ represents the best object in the same membrane o , and $\|D\|$ is the size of the data space given by $\|D\| = \sqrt{\sum_{j=1}^T (x_j^{\max} - x_j^{\min})^2}$ with $T = Kd$. The best neighbor $u_p^{\text{best}}(t)$ of u_p has the greatest FER value among all neighbors in the neighborhood.

4.2.2. The Communication Rules. The communication rules in the PSO-CP realize the exchange and sharing of the global best objects among elementary membranes 2 to q and nonelementary membrane 1 as well as between nonelementary membrane 1 and skin membrane 0. This communication only exists between children membranes and the parent membrane. In order to enhance the global searching ability of the objects, the PSO-CP has two types of communication rules, i.e., the communication rules in the elementary membranes and the communication rules in the nonelementary membranes.

- (1) $R'_o = \{u_o^{\text{gbest}}(t) \longrightarrow (u_{p_o}^{\text{best}}(t), in_1)\}$, for $o = 2, 3, \dots, q$

A copy of the global best object $u_o^{\text{gbest}}(t)$ in elementary membrane o is evolved to be the local best object $u_{p_o}^{\text{best}}(t)$ and is sent to nonelementary membrane 1 at iteration t . Note that the global best object $u_o^{\text{gbest}}(t)$ still stays in elementary membrane o . Therefore, nonelementary membrane 1 only contains $q-1$ objects from the $q-1$ elementary membranes at iteration t . Nonelementary membrane 1 will select the best object among these local best objects from the elementary membranes as the global best object $u_1^{\text{gbest}}(t)$.

- (2) $R'_1 = \{u_{p_o}^{\text{best}}(t) \longrightarrow (u_r^{\text{best}}(t), in_{o+1})\}$, for $o = 2, 3, \dots, q-1$ and $r = 1, 2, \dots, m$

A copy of the local best object $u_{p_o}^{\text{best}}(t)$ in nonelementary membrane 1 is evolved to be the local best $u_r^{\text{best}}(t)$ of u_r and is sent to elementary membrane $o+1$ at iteration t . The selection of u_r is based on a random strategy in membrane $o+1$. Specially, $R'_1 = \{u_{p_q}^{\text{best}}(t) \longrightarrow (u_r^{\text{best}}(t), in_2)\}$, a copy of the local best object $u_{p_q}^{\text{best}}(t)$ is evolved to be the local best $u_r^{\text{best}}(t)$ of u_r in elementary membrane 2, while the selection of u_r is based on a random strategy. At the same time, membrane 1 will transport a copy of the global best object $u_1^{\text{gbest}}(t)$ to the skin membrane 0 and evolve it to the local best object $u_0^{\text{best}}(t)$, i.e., $R'_1 = \{u_1^{\text{gbest}}(t) \longrightarrow (u_0^{\text{best}}(t), in_0)\}$. If the local best object $u_0^{\text{best}}(t)$ at iteration t is better than the global best object $u_0^{\text{gbest}}(t+1)$ at iteration $t-1$ in the skin membrane 0, the local best object $u_0^{\text{best}}(t)$ will become the global best object $u_0^{\text{gbest}}(t)$ at iteration t . Thus, the skin membrane 0 always holds the global best object of the whole membrane system at iteration t and will output it to the environment at the end of the evolution and computation process. In addition, the local best object $u_0^{\text{best}}(0)$ from nonelementary membrane 1 will be placed into the skin membrane 0 as the best object $u_0^{\text{gbest}}(0)$ at the beginning of the process. Figure 3 graphically depicts the communication rules in the PSO-CP. The red arrows represent the directions of the communicated information. The exchange and transmission of global information is achieved by the execution of communication rules between nonelementary and elementary membranes.

4.3. Creation and Dissolution Rules for Membranes. Membrane rules form the evolution mechanism of membrane systems, which are different from the traditional evolution and communication rules of the objects. The PSO-CP uses two types of membrane rules: membrane creation and dissolution rules [50]. The membrane rules only change the elementary membranes, while the nonelementary and skin membranes always stay the same in the evolution and computation process.

4.3.1. Membrane Creation Rules. The membrane creation rules are of the form $[u]_o \longrightarrow [u]_o, [u]_{o_1}, \dots, [u]_{o_{sn}}$, where $u \in O$. The creation rules are executed when the stagnation condition is detected. The stagnation condition is when the global best in the elementary membranes cannot be further improved for *limit* iterations. When the membrane creation rules are executed, a subsystem of membrane o is created. This subsystem consists of sn membranes, each of which is independent of the others.

When a new subsystem is created, all the objects in the current elementary membrane o are copied into each of the membranes in the subsystem. The local evolution rules for objects are adopted in membranes o_1 to o_{sn} of the subsystem. After one evolution, each membrane will transport a copy of the global best object $u_{o_g}^{\text{gbest}}$, for $g = 1, 2, \dots, sn$, to

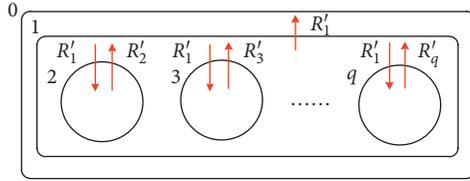


FIGURE 3: The communication rules in the PSO-CP.

nonelementary membrane 1. Nonelementary membrane 1 will then select the best object v_o^{gbest} among the global best objects u_o^{gbest} in the membranes of the subsystem and the global best object u_o^{gbest} in elementary membrane o and retain the best membrane containing the global best object to continue the following evolution and computation process.

Figure 4 shows an example of membrane creation. At iteration t , a new subsystem of elementary membrane 3 is created when the global best object cannot be further improved for *limit* iterations, and this subsystem consists of sn membranes. All the objects in membrane 3 are copied to the membranes of this subsystem. The local evolution rules for objects are used to evolve the objects in each membrane in the subsystem. At the same time, elementary membrane 3 continues to evolve. After one evolution, each membrane in the subsystem will transport a copy of the global best object $u_{3_g}^{\text{gbest}}(t)$, for $g = 1, 2, \dots, sn$, to nonelementary membrane 1. Nonelementary membrane 1 selects the global best object $v_3^{\text{gbest}}(t+1)$ among the global best objects $u_{3_g}^{\text{gbest}}(t+1)$ in the membranes of the subsystem and the global best object $u_3^{\text{gbest}}(t+1)$ in elementary membrane 3. The membrane containing the global best object $v_3^{\text{gbest}}(t+1)$ will be kept and will be used to replace elementary membrane 3.

4.3.2. Membrane Dissolution Rules. During the evolution and computation process of the PSO-CP, the membranes will be dissolved through the membrane dissolution rules, but dissolution happens only on elementary membranes and the ones in their corresponding subsystems. The PSO-CP destroys a subsystem by the membrane dissolution rules when the best object in a subsystem or in the corresponding elementary membrane is found after one evolution. The membrane dissolution rules are of the form $[u] \rightarrow \lambda$.

Membrane dissolution happens in membrane o_g of the subsystem and the corresponding elementary membrane o , for $g = 1, 2, \dots, sn$ and $o = 2, 3, \dots, q$. After one evolution, each membrane in the subsystem will transport a copy of the global best object $u_{o_g}^{\text{gbest}}$ to nonelementary membrane 1. If the global best object v_o^{gbest} is from membrane o_g in the subsystem and is better than the global best object u_o^{gbest} of the corresponding elementary membrane o and other global best objects $u_{o_{g'}}^{\text{gbest}}$ in the membranes of the subsystem, elementary membrane o and other membranes $o_{g'} (g' \neq g)$ in the subsystem will be dissolved through the membrane dissolution rules. Membrane o_g in the subsystem will replace elementary membrane o and will continue to perform the subsequent evolutions and computations with the same evolution and communication rules in elementary membrane o . Otherwise, if the global best object v_o^{gbest} is from the

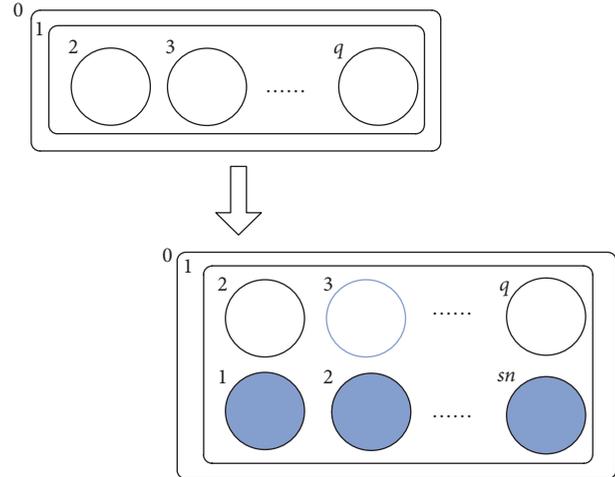


FIGURE 4: An example of membrane creation.

global best object u_o^{gbest} in elementary membrane o , all the membranes o_1 to o_{sn} in the subsystem will be dissolved.

Figure 5 shows an example of membrane dissolution. At iteration t , the subsystem of elementary membrane 3 is generated, which contains sn membranes. After one evolution, each membrane will transport a copy of the global best object $u_{3_g}^{\text{gbest}}(t+1)$, for $g = 1, 2, \dots, sn$, to nonelementary membrane 1. Nonelementary membrane 1 selects the global best object $v_3^{\text{gbest}}(t+1)$ among the global best objects $u_{3_g}^{\text{gbest}}(t+1)$ in the membranes of the subsystem and the global best object $u_3^{\text{gbest}}(t+1)$ in elementary membrane 3. Assume that the global best object $v_3^{\text{gbest}}(t+1)$ comes from membrane 1 in the subsystem. Therefore, membrane 1 in the subsystem replaces elementary membrane 3 and elementary membrane 3 and all other membranes $3_{g'} (g' \neq 1)$ in the subsystem are dissolved. After replacing elementary membrane 3, membrane 1 in the subsystem is the new elementary membrane 3 which will continue to perform the subsequent evolutions and computations.

4.4. Halting and Output. The PSO-CP is a parallel computing system, all elementary membranes and their corresponding subsystems work in parallel, and each membrane including elementary membranes and membranes in their subsystems are parallel computing units. The extended clustering membrane system starts running from the initial membrane structure with $q - 1$ elementary membranes containing the initial objects. Each object represents a set of cluster centers. These objects will be evolved based on the basic evolution rules. The elementary membranes and nonelementary membranes interact through the communication rules, and the global best object in nonelementary membrane 1 will be transported to the skin membrane 0. A subsystem will be generated based on the membrane creation rules when an elementary membrane is trapped into a local optimum, and the local evolution rules are used to evolve the objects in membranes of the subsystem. After one evolution, the membrane with the best object among the membranes in the subsystem and the corresponding

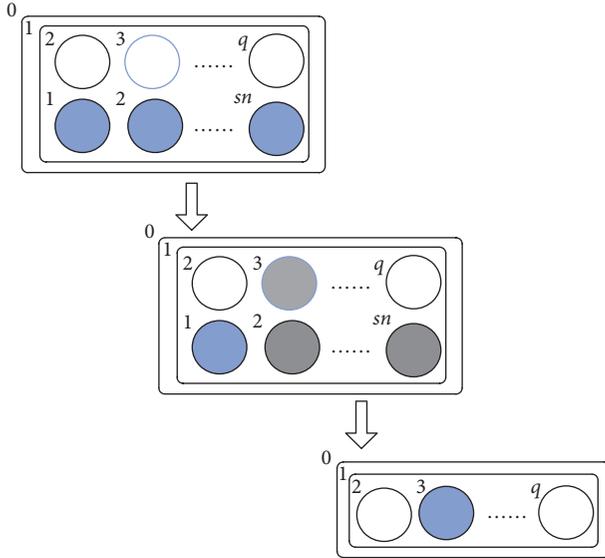


FIGURE 5: An example of membrane dissolution.

elementary membrane will be kept and others will be dissolved based on the membrane dissolution rules. The evolution and communication rules for objects and the creation and dissolution rules for membranes will execute iteratively during the evolution and computation process. These computing tasks are performed iteratively. The extended clustering membrane system will continue to execute until the halting condition is satisfied, which is the maximum number of iterations has been reached. When the system halts, the last global best object stored in skin membrane 0 is output to the environment and the set of cluster centers of the clustering problem is regarded as the final computed result.

4.5. Complexity Analysis. In this subsection, the complexity of the PSO-CP is analyzed. As defined earlier, N is the number of data points in the datasets, m represents the number of objects in an elementary membrane, q represents the number of elementary membranes in the system, and t_{\max} represents the maximum number of iterations. In the initialization process, the local best and global best in each of the elementary membranes 2 to q need to be found in maximal parallel. The complexity of initialization is then $O(m)$. The basic evolution rules in the elementary membranes are executed in parallel. The time needed by executing a basic evolution rule for an object is N . Hence, the time needed by one evolution in an elementary membrane is mN . The communication rules between elementary membranes and nonelementary membranes are also executed in parallel, and the time needed by executing a communication rule is 2. The time needed by a membrane in a subsystem is $m(m + N)$. Thus, the time needed by the membranes in a subsystem and the corresponding elementary membrane is $m(m + N)$. As a result, the time needed by each iteration is $m(m + N) + 2$. The total time needed by the PSO-CP is $m + t_{\max}(m(m + N) + 2)$. Because $m \ll N$, the complexity of the PSO-CP can be simplified to $O(t_{\max}mN)$.

5. Experimental Results and Analysis

Computational experiments are conducted to evaluate the effectiveness of the PSO-CP. The datasets used in this study are introduced first. Four artificial datasets [51] are then used to tune the parameters in the PSO-CP. The clustering performance of the PSO-CP is compared with those of currently existing approaches using ten test datasets [52]. All clustering methods, including the PSO-CP, are implemented using MATLAB 2016b, and all the experiments are conducted on a Dell desktop computer with an Intel 4.00 GHz i7-8550U processor and 8 GB of RAM in the Windows 10 environment.

5.1. Datasets. Four artificial datasets and ten test datasets are used in the experiments. The four artificial datasets are used to tune the parameters of the PSO-CP, and the ten test datasets are used to test the clustering performance of the PSO-CP as compared with those of five other clustering approaches. The ten test datasets including three artificial datasets and seven real-life datasets have been used by researchers as benchmarks to test their clustering approaches. These datasets are briefly described below.

The seven artificial datasets, *Data_5_2*, *Size_5*, *Square4*, *LineBlobs*, *Data_4_3*, *Data_9_2*, and *Square1*, are manually generated and have been used in the existing literature. The seven real-life datasets, *Iris*, *Newthyroid*, *Seeds*, *Yeast*, *Glass*, *Wine*, and *Lung Cancer*, are from the UCI Machine Learning Repository. More details about these datasets are presented in Tables 1 and 2.

The *Lung Cancer* dataset is high dimensional. It contains 32 data points, has 56 independent features, and describes three types of lung cancers. The *Yeast* dataset consists of 1484 data points and has 8 features. The *Iris* and *Glass* datasets are not linearly separable in the Euclidean space. These datasets with different characteristics in shape, size, compactness, and symmetry are used to evaluate the performance of the PSO-CP quantitatively.

5.2. Parameter Settings. The number of membranes in the subsystems and the number of elementary membranes play important roles in the performance of the PSO-CP. The values of these parameters have critical influences on the performance of the PSO-CP in solving clustering problems. This section focuses on checking the influences of two parameters, i.e., the number of elementary membranes $q - 1$ and the number of membranes sn in the subsystems, with four artificial datasets.

5.2.1. Number of Elementary Membranes. In order to evaluate the effects of the number of elementary membranes on clustering performance, the PSO-CP with two different degrees, i.e., 2 and 5, are used to find the optimal clustering centers for these datasets, using the fitness function to measure clustering quality [34]. The PSO-CP with each degree ran 30 times on each dataset and the mean (Mean) and standard deviation (S.D.) of the fitness values are

TABLE 1: Properties of the artificial datasets.

Datasets	Data	Feature	Class
<i>Data_5_2</i>	250	2	5
<i>Size_5</i>	1000	2	4
<i>Square4</i>	1000	2	4
<i>LineBlobs</i>	266	3	4
<i>Data_4_3</i>	400	3	4
<i>Data_9_2</i>	900	2	9
<i>Square1</i>	1000	2	4

TABLE 2: Properties of the real-life datasets.

Datasets	Data	Feature	Class
<i>Iris</i>	150	4	3
<i>Newthyroid</i>	215	5	3
<i>Seeds</i>	210	7	3
<i>Yeast</i>	1484	8	10
<i>Glass</i>	214	9	6
<i>Wine</i>	178	13	3
<i>Lung Cancer</i>	32	56	3

reported. The S.D. shows the robustness and stability of the clustering technique.

Parameters of the PSO-CP not tested in this experiment are kept at the same values for the fairness in the comparisons. In the PSO-CP, the number of objects may affect the quality of the clustering results. In order to avoid the influence of the number of objects, the number of objects in the system is set to the same value, i.e., $M = 100$. The maximum number of iterations is set to $t_{\max} = 100$, and the values of the learning factors in the basic and local evolution rules are set to $c_1 = c_2 = 2$. The number of membranes in the subsystem is set to $sn = 6$. The values of the lower and upper limits of the inertia weight are set to $w_{\min} = 0.4$ and $w_{\max} = 1.2$, respectively. For the detection of the stagnation condition of an elementary membrane, the number of iterations is set to $limit = 2$. The values of Mean and S.D. of the fitness function obtained by the PSO-CP with two degrees are reported in Table 3.

Table 3 gives the results of the PSO-CP with different degrees on the four artificial datasets. The best values of Mean and S.D. for each dataset are highlighted. The results show that the PSO-CP obtained smaller Mean and smaller S.D. when $q - 1 = 5$ elementary membranes are used. These results also show the PSO-CP with $q - 1 = 5$ has better robustness. Figure 6 gives more details about the clustering performance of the PSO-CP on two datasets, i.e., *Data_5_2* and *Size_5*. The clustering performance of the PSO-CP may remarkably improve with the increase in the number of elementary membranes.

5.2.2. Number of Membranes in the Subsystem. The number of membranes in the subsystem also has an important influence on the performance of the PSO-CP. In order to tune the number of membranes, the clustering performances of the PSO-CP with $sn = 6, 10$, and 14 are examined [12]. The parameters of the PSO-CP which are not tested in this

TABLE 3: Performance of the PSO-CP with different degrees measured by the values of Mean and S.D. of the fitness function.

Datasets	$q = 3$		$q = 6$	
	Mean	S.D.	Mean	S.D.
<i>Data_5_2</i>	326.7537	0.5323	326.5707	0.1594
<i>Size_5</i>	2499.3796	15.1835	2499.2712	14.7658
<i>Square4</i>	2367.7321	0.2262	2367.7097	0.2652
<i>LineBlobs</i>	20.6297	0.0018	20.6294	0.0016

experiment are kept at the same values. The values of Mean and S.D. of the fitness function are presented in Table 4.

The PSO-CP with each different number of membranes in the subsystem ran 30 times on each dataset. The Mean and S.D. of the values of the fitness function for each number of membranes in the subsystem on each dataset are presented in Table 4. The PSO-CP shows best performance on these datasets when $sn = 10$. Figure 7 gives the boxplots of the values of the fitness function for each number of membranes in the subsystem for two datasets *Data_5_2* and *Size_5*. Considering results for all four datasets, the PSO-CP with $sn = 10$ has the best overall performance.

5.3. Comparison with Other Clustering Approaches. To evaluate the effectiveness of the PSO-CP, its performance is compared with those of the standard particle swarm optimization (PSO) [49], fitness-distance ratio-particle swarm optimization (FDR-PSO) [53], fitness-Euclidean ratio-particle swarm optimization (FER-PSO) [42], starling particle swarm optimization (SPSO) [12], and environment particle swarm optimization (EPSO) [13] on the ten test datasets. Although there are many other advanced PSO clustering approaches, the ones used for comparison are the major references for the development of, and are more relevant to, the PSO-CP. The neighbor of the particle with the best FDR or FER value is selected in the FDR-PSO and FER-PSO approaches to replace the local best for velocity updating. A mechanism is used to lead the search out of a local optimum when stagnation occurs in the SPSO approach, and the implementation of the environment factor in the EPSO approach is similar to the active membranes in this study.

Table 5 reports the parameter values of all the comparative clustering approaches used in the experiments. The information of neighbors is used to guide the search direction of the particle, and the number of neighbors has important influence on the performance of the SPSO approach. The SPSO approach copies the current population to multipopulations when the cumulative number of iterations reaches a previously set threshold, called stagnant limit meaning that the particle in the population is trapped into a local optimum. The number of subpopulations is a decisive factor of the convergent rate in the approach. This operation of population is similar to the creation and dissolution of membranes in the PSO-CP. The velocity control of FER-PSO is an adjustable parameter to balance the previous velocity and the current velocity of the particle, and the learning factor controls the influence of the environmental factor in EPSO. The values of all these adjustable parameters in these

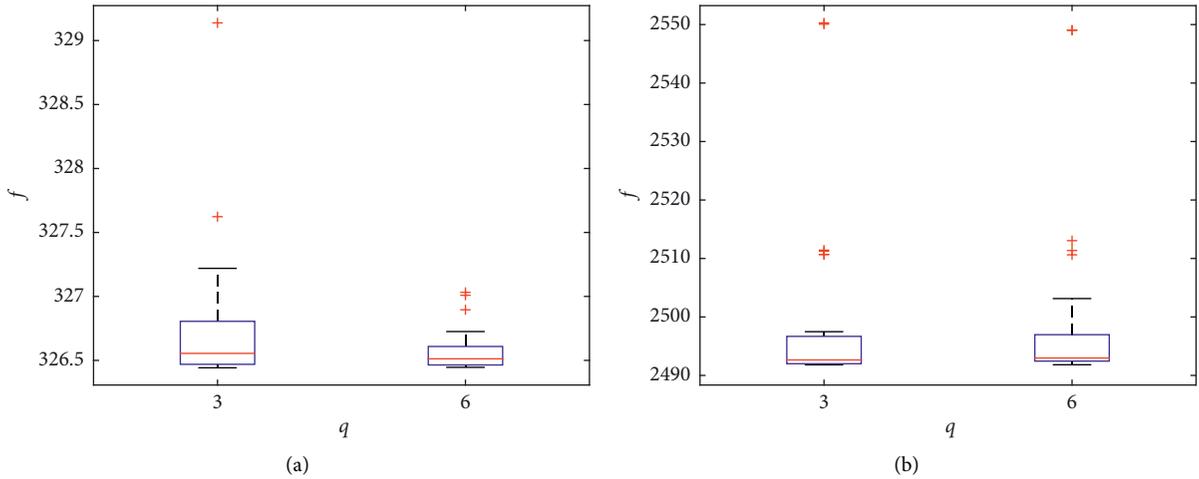


FIGURE 6: Boxplots of the values of the fitness function for 30 runs by the PSO-CP with different degrees: (a) *Data_5_2* and (b) *Size_5*.

TABLE 4: Performance of the PSO-CP with different number of membranes in the subsystems measured by the values of Mean and S.D. of the fitness function.

Datasets	$sn = 6$		$sn = 10$		$sn = 14$	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
<i>Data_5_2</i>	326.5707	0.1594	326.5702	0.1210	326.7454	0.4251
<i>Size_5</i>	2499.2712	14.7658	2496.0993	11.2346	2497.4146	11.9762
<i>Square4</i>	2367.7097	0.2652	2367.6292	0.1411	2367.8010	0.2982
<i>LineBlobs</i>	20.6294	0.0016	16.7426	0.0362	16.7937	0.0391

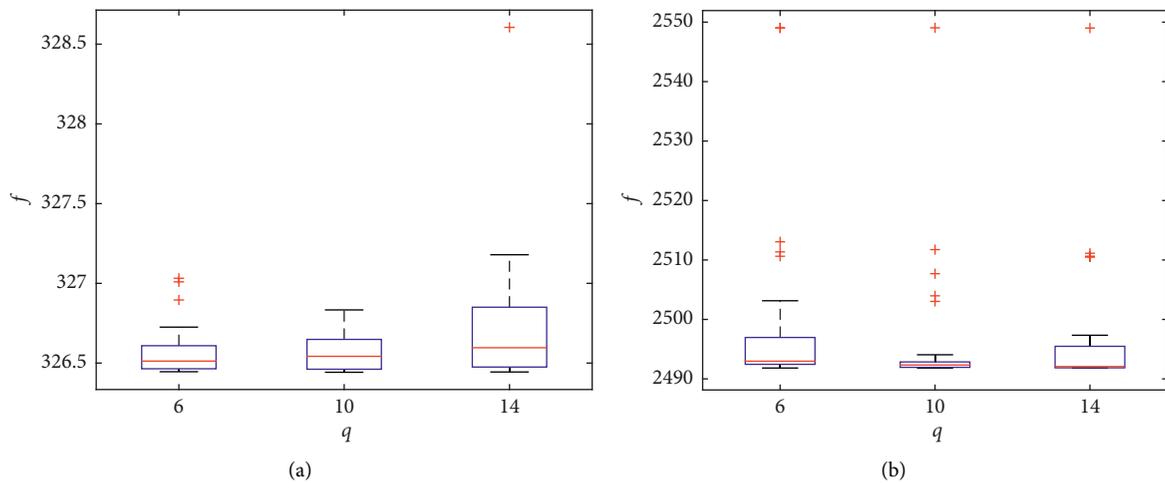


FIGURE 7: Boxplots of the values of the fitness function for 30 runs of the PSO-CP with different number of membranes in the subsystem: (a) *Data_5_2* and (b) *Size_5*.

approaches are the best values reported in the respective publications.

Each clustering approach, including the PSO-CP, ran 50 times for each dataset. Simple statistics including the worst value (Worst), the best value (Best), the Mean, and the S.D. of the fitness function are used as the evaluation criteria. The experimental environment is the same for all these comparative clustering approaches.

Figure 8 shows the convergence of these clustering approaches on the ten test datasets for typical runs of these approaches. The fitness value obtained by the PSO-CP 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, FDR-PSO, and FER-PSO decrease slowly at the beginning of the evolution process and do not apparently have better convergence performance

TABLE 5: Parameter settings of the comparative clustering approaches used in the experiments.

Parameters	PSO	FDR-PSO	FER-PSO	SPSO	EPSO	PSO-CP
Population (M)	100	100	100	100	100	100
t_{\max}	100	100	100	100	100	100
c_1, c_2	2, 2	2, 2	0.68, 0.68	2, 2	2, 2	2, 2
c_3	N	N	N	N	2	N
r_1, r_2	(0, 1)	(0, 1)	(0.2, 0.5)	(0, 1)	(0, 1)	(0, 1)
(w_{\min}, w_{\max})	1	1	1.37	(0.2, 0.4)	(0.4, 0.6)	(0.4, 1.2)
Neighbors	N	N	N	7	N	N
Subsystem (Subpopulations)	N	N	N	14	N	10
Velocity control	N	N	4.1	N	N	N
Stagnant limit	N	N	N	2	N	2
Degrees	N	N	N	N	N	5

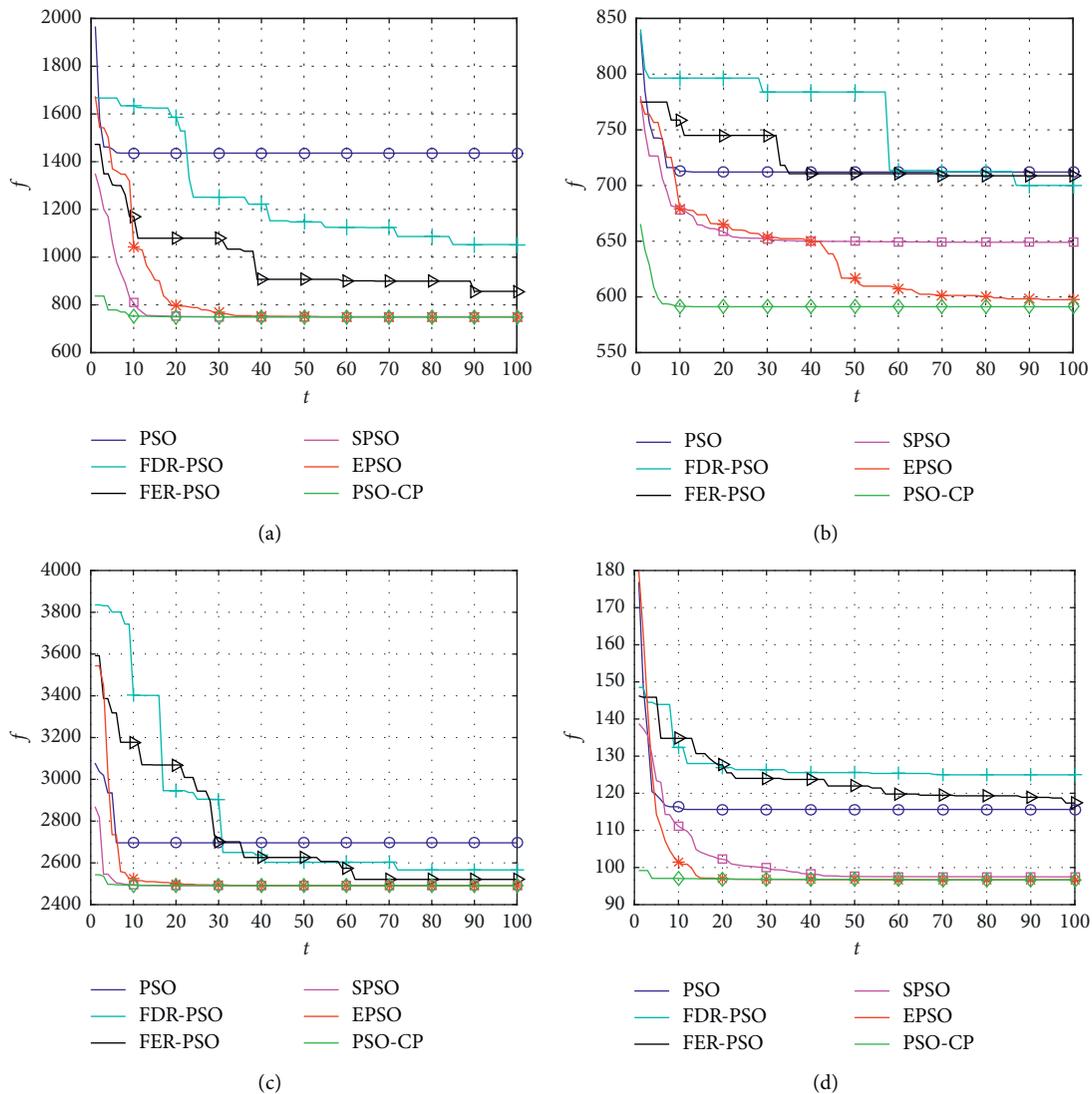


FIGURE 8: Continued.

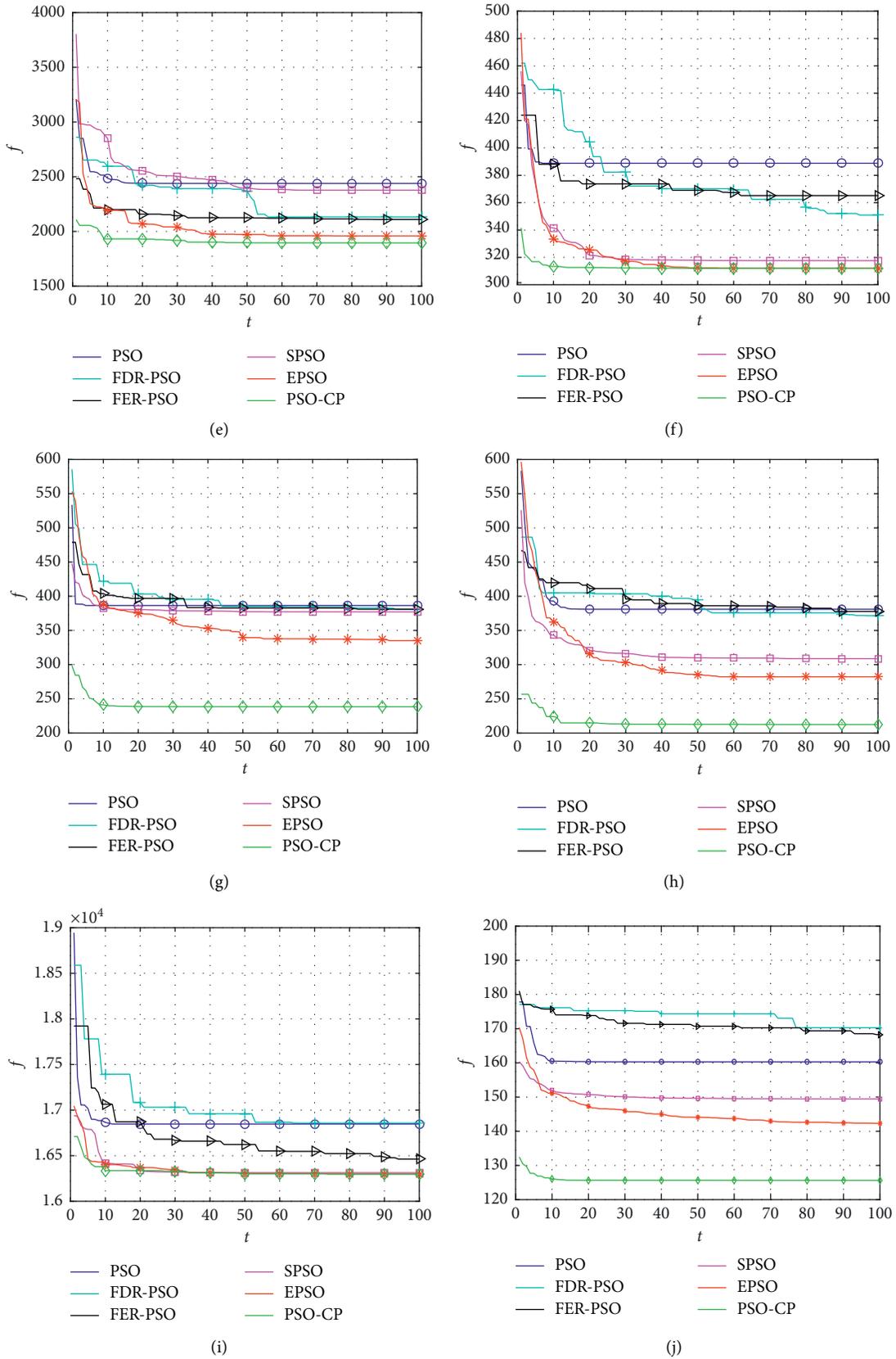


FIGURE 8: Convergence of the approaches on the ten test datasets in terms of fitness values. (a) *Data_4_3*. (b) *Data_9_2*. (c) *Square1*. (d) *Iris*. (e) *Newthroid*. (f) *Seeds*. (g) *Yeast*. (h) *Glass*. (i) *Wine*. (j) *Lung Cancer*.

TABLE 6: Performances of the comparative clustering approaches on the ten test datasets measured by the values of the fitness function.

Datasets	Parameters	Clustering Approaches					
		PSO	FDR-PSO	FER-PSO	SPSO	EPSO	PSO-CP
<i>Data_4_3</i>	Best	826.6171	819.7949	898.7706	749.5980	749.6758	749.6927
	Worst	1850.0014	1482.8549	1360.0130	1288.8033	751.8741	749.8452
	Mean	1320.8028	1137.2361	1074.4874	1067.7040	750.4022	749.7642
	S.D.	286.2578	208.0353	135.2255	259.7331	0.5539	0.0444
<i>Data_9_2</i>	Best	701.0206	666.3878	672.0371	590.8895	592.3423	590.9903
	Worst	821.5692	769.4857	746.3306	656.6909	647.1212	591.2978
	Mean	772.3889	724.7441	709.6785	610.6185	605.4559	591.1481
	S.D.	28.2783	24.5240	24.8687	24.73122	17.5056	0.0668
<i>Square1</i>	Best	2505.3057	2502.2471	2500.4159	2491.2407	2491.3787	2491.2439
	Worst	3640.3065	2675.2805	2576.7491	2491.2410	2491.2379	2492.9382
	Mean	2877.8554	2554.4594	2528.9649	2491.2410	2491.6646	2491.4117
	S.D.	356.5662	45.1811	22.6925	3.42E-12	0.2295	0.2239
<i>Iris</i>	Best	98.7842	99.5456	100.9861	96.6555	96.6751	96.7218
	Worst	158.0786	139.0893	129.3110	127.6677	97.4857	96.9734
	Mean	122.9807	120.0652	111.1305	98.4060	96.9072	96.7981
	S.D.	17.9146	9.8442	8.9548	6.8960	0.2477	0.0606
<i>Newthroid</i>	Best	1960.8256	1964.3799	1971.9836	1939.0783	1912.8158	1884.2624
	Worst	3335.1916	2536.9953	2387.9253	2425.7106	2086.3923	1913.2869
	Mean	2506.2474	2236.3322	2126.8844	2174.3161	1980.1139	1901.4620
	S.D.	327.3508	151.6784	106.9792	138.9515	43.2650	6.3795
<i>Seeds</i>	Best	350.0677	338.9096	338.8187	313.8338	312.0655	311.9079
	Worst	488.4095	430.6464	430.1311	330.3819	319.5618	312.2222
	Mean	408.3179	365.8090	363.9724	320.2067	314.4442	312.0261
	S.D.	38.4531	22.1887	20.6664	4.3349	1.8426	0.0693
<i>Yeast</i>	Best	377.4794	379.6469	369.5542	334.8136	310.6105	236.4264
	Worst	398.8870	385.1700	383.2536	380.2217	369.7602	242.9336
	Mean	383.7480	382.2182	380.7361	366.4061	346.1302	239.2609
	S.D.	5.0879	1.8023	3.0165	12.5515	18.2818	1.9088
<i>Glass</i>	Best	323.8897	313.9315	320.6142	295.7293	268.7450	212.4726
	Worst	505.9255	408.4166	396.4361	374.6702	298.358	213.3599
	Mean	392.0423	363.3179	351.4392	316.4029	285.3853	212.9749
	S.D.	38.8437	31.4624	21.2513	19.4952	8.63419	0.2531
<i>Wine</i>	Best	16478.4314	16425.3324	16394.2432	16303.2915	16297.6997	16294.4696
	Worst	18718.6492	16863.3335	16765.0099	16360.8555	16321.6502	16311.2132
	Mean	17242.4999	16612.5659	16530.1668	16326.2498	16308.3558	16301.2990
	S.D.	551.5428	134.3416	109.0889	18.4980	7.9560	4.48774
<i>Lung Cancer</i>	Best	153.9672	164.5567	161.0568	147.5225	135.4626	125.6519
	Worst	169.6352	178.3152	172.3457	152.6790	141.9717	125.6683
	Mean	160.3950	170.2530	167.3035	149.1543	138.4026	125.6621
	S.D.	3.6903	3.8259	3.2550	1.3824	1.8706	0.0045

than other approaches. Because the neighbor with the best FDR and FER values is used to replace the local best of the particle, the convergence of FDR-PSO and FER-PSO is not as fast as that of PSO. Although SPSO and EPSO show better performance than the above clustering approaches, they are also easily trapped into local optima, as shown in parts (b), (g), (h), and (j) of Figure 8. Therefore, the PSO-CP has better convergence speed and higher clustering quality than the comparative approaches for all these datasets, as shown in Figure 8.

Simple statistics of the fitness function values of these clustering approaches on these datasets are reported in Table 6. Results in Table 6 show that the PSO-CP has the overall best performance on these ten test datasets. Because of the characteristics of the test datasets, some

clustering approaches performed better on some specific datasets with smaller mean values, but the performance of the PSO-CP on these test datasets is considered comparable. Compared with other clustering approaches, the PSO-CP is more robust with smaller values of S.D. of the fitness function values, and its performance is more stable than PSO through the use of the extended cell-like P system.

To investigate the performance of the PSO-CP, the average values of the fitness function are compared with those of the other clustering approaches and the Friedman test is used in the comparison. The null hypothesis is that all the clustering approaches in this experiment have the same performance for any one dataset. Mathematically, the Friedman test works as follows [54].

In the Friedman test, the ten test datasets are treated as a random sample and each clustering approach is considered as a treatment. The average fitness values of the clustering approaches on each dataset are ranked from the largest to the smallest [55]. The rank of clustering approach j on clustering problem i is denoted by r_{ij} . The mean of these ranks is $(1/2)(p+1)$, in this case 3.5, where p is the number of treatments. The Friedman test statistic χ_r^2 is given in the following form:

$$\chi_r^2 = \frac{12}{np(p+1)} \sum_{p=1}^p \left(\sum_{i=1}^n r_{ij} \right)^2 - 3n(p+1), \quad (14)$$

where n is the number of rows, i.e., datasets, 10 in this case. The Friedman test statistic follows a chi-squared distribution with $p-1$ degrees of freedom.

The ranks of the fitness values obtained by the clustering approaches for each dataset are presented in Table 7. The PSO-CP is ranked highest, i.e., nine out of the ten test datasets. The Friedman test statistic χ_r^2 is computed using Table 7, and the result is $\chi_r^2 = 46.2286$. With $p-1 = 5$ degrees of freedom, the critical value is $\chi^2 = 11.07$ at the significance level $\alpha = 0.05$. Hence, the conclusion of the Friedman test is to reject the null hypothesis, i.e., the treatment levels are significantly different. In this case, the different clustering approaches obtained significantly different fitness function values.

In clustering problems, the F -measure is sometimes used to measure the quality of clustering [35]. Each data point in a dataset belongs to a specific class, i.e., has a specific label, in reality although the label is usually unknown for clustering problems. Let \mathbf{b}_l represent class l in reality and \mathbf{c}_k represent cluster k obtained by a clustering approach, for $l, k = 1, 2, \dots, K$. The number of data points belonging to \mathbf{b}_l is denoted by $|\mathbf{b}_l|$, the number of data points belonging to \mathbf{c}_k is denoted by $|\mathbf{c}_k|$, and the number of the data points belonging to both \mathbf{b}_l and \mathbf{c}_k is denoted by $|\mathbf{b}_l \cap \mathbf{c}_k|$. The precision of class l and cluster k is defined as follows:

$$P(\mathbf{b}_l, \mathbf{c}_k) = \frac{|\mathbf{b}_l \cap \mathbf{c}_k|}{|\mathbf{c}_k|}. \quad (15)$$

The recall of class l and cluster k is defined as follows:

$$R(\mathbf{b}_l, \mathbf{c}_k) = \frac{|\mathbf{b}_l \cap \mathbf{c}_k|}{|\mathbf{b}_l|}. \quad (16)$$

The F -measure of class l and cluster k is given in the following form:

$$F(\mathbf{b}_l, \mathbf{c}_k) = \frac{2P(\mathbf{b}_l, \mathbf{c}_k)R(\mathbf{b}_l, \mathbf{c}_k)}{P(\mathbf{b}_l, \mathbf{c}_k) + R(\mathbf{b}_l, \mathbf{c}_k)}. \quad (17)$$

The clustering results of a good clustering approach should be close to the actual classes in the dataset [56]. The overall F -measure of the clustering results is given in the following form:

$$F = \sum_{l=1}^K \frac{|\mathbf{b}_l|}{N} F(\mathbf{b}_l) = \sum_{l=1}^K \frac{|\mathbf{b}_l|}{N} \max_{1 \leq k \leq K} F(\mathbf{b}_l, \mathbf{c}_k). \quad (18)$$

TABLE 7: Ranks of the average values for the six clustering methods (Mean) and computation of the Friedman test statistic.

Datasets	PSO	FDR-PSO	FER-PSO	SPSO	EPSO	PSO-CP
<i>Data_4_3</i>	6	5	4	3	2	1
<i>Data_9_2</i>	6	5	4	3	2	1
<i>Square1</i>	6	5	4	1	3	2
<i>Iris</i>	6	5	4	3	2	1
<i>Newthyroid</i>	6	5	3	4	2	1
<i>Seeds</i>	6	5	4	3	2	1
<i>Yeast</i>	6	5	4	3	2	1
<i>Glass</i>	6	5	4	3	2	1
<i>Wine</i>	6	5	4	3	2	1
<i>Lung Cancer</i>	4	6	5	3	2	1
Total rank	58	51	40	29	21	11
Average rank	5.8	5.1	4.0	2.9	2.1	1.1
Deviation	2.3	1.6	0.5	-0.6	-1.4	-2.4

When clustering approaches are compared, the approach with a larger value of the F -measure is more effective.

Table 8 provides the Mean and S.D. of the values of the F -measure of the comparative clustering approaches on the ten test datasets. As the results in Table 8 show, the PSO-CP clearly has the best overall performance according to the F -measure among all these clustering approaches.

The classification of a data point is correct or accurate if it is clustered into the right class or cluster [56]. Therefore, the classification rate represented by A , also called clustering accuracy, for the test datasets is also used to evaluate the performance of the clustering approaches. The classification rate of a dataset is defined as the proportion of correctly classified data points in a dataset, as shown in the following equation:

$$A = \frac{E}{N}, \quad (19)$$

where E is the number of correctly classified data points. Table 9 provides the classification rates of the comparative clustering approaches on the ten test datasets. Although the PSO-CP obtained a classification rate lower than that of FER-PSO on the *Square1* dataset, it obtained better classification rates on all other test datasets than the other clustering approaches. Overall, the PSO-CP has the highest means of classification rates among the six comparative clustering techniques.

The Friedman test is also applied to the Means of the classification rates. The computed Friedman test statistic is $\chi_r^2 = 50.4$. With $p-1 = 5$ degrees of freedom, the critical value is $\chi^2 = 11.07$ at the significance level $\alpha = 0.05$. Therefore, these clustering approaches obtained significantly different classification rates.

Compared with other improved PSO procedures, the PSO-CP has better values of the F -measure and better classification rates. Therefore, the extended cell-like P system helps the PSO-CP improve its clustering performance, and the introduction of membrane systems gives a new way for PSO to solve clustering problems.

TABLE 8: *F*-measure of the ten test datasets obtained by the comparative clustering approaches.

Datasets	Statistics	Clustering Approaches					
		PSO	FDR-PSO	FER-PSO	SPSO	EPSO	PSO-CP
<i>Data_4_3</i>	Mean	0.9489	0.9000	0.7753	0.8500	1	1
	S.D.	0.1025	0.1257	0.1381	0.1257	0	0
<i>Data_9_2</i>	Mean	0.7951	0.7924	0.7174	0.8853	0.9015	0.9209
	S.D.	0.0460	0.0454	0.0560	0.0510	0.0446	0.0024
<i>Square1</i>	Mean	0.9893	0.9886	0.9734	0.9890	0.9890	0.9890
	S.D.	0.0014	0.0021	0.0223	4.56E – 16	4.56E – 16	4.56E – 16
<i>Iris</i>	Mean	0.8383	0.7833	0.7887	0.8903	0.8977	0.9003
	S.D.	0.0918	0.0997	0.1111	0.0534	0.0032	0.0070
<i>Newthyroid</i>	Mean	0.7437	0.7386	0.7570	0.7630	0.7616	0.7988
	S.D.	0.0417	0.0320	0.0370	0.0378	0.0177	0.0122
<i>Seeds</i>	Mean	0.8721	0.8671	0.8069	0.8950	0.8945	0.8952
	S.D.	0.0541	0.0534	0.1017	0.0063	0.0017	1.14E – 16
<i>Yeast</i>	Mean	0.3188	0.3166	0.3228	0.3437	0.3748	0.5171
	S.D.	0.0075	0.0046	0.0105	0.0217	0.0204	0.0158
<i>Glass</i>	Mean	0.4402	0.4334	0.4308	0.4717	0.5061	0.5843
	S.D.	0.0391	0.0496	0.0460	0.0285	0.0149	0.0010
<i>Wine</i>	Mean	0.7096	0.7104	0.7084	0.7143	0.7166	0.7169
	S.D.	0.0041	0.0050	0.0075	0.0052	0.0039	0.0038
<i>Lung Cancer</i>	Mean	0.4172	0.4266	0.4594	0.4641	0.5312	0.5625
	S.D.	0.0233	0.0355	0.0487	0.0659	0.0641	1.12E – 17

TABLE 9: Classification rates of the ten test datasets obtained by the comparative clustering approaches.

Datasets	Statistics	Clustering Approaches					
		PSO	FDR-PSO	FER-PSO	SPSO	EPSO	PSO-CP
<i>Data_4_3</i>	Mean	0.6746	0.8618	0.9230	0.7732	1	1
	S.D.	0.2019	0.1830	0.1640	0.1948	0	0
<i>Data_9_2</i>	Mean	0.7157	0.7873	0.7918	0.8820	0.8993	0.9211
	S.D.	0.0530	0.0474	0.0461	0.0560	0.0496	0.0024
<i>Square1</i>	Mean	0.9731	0.9886	0.9893	0.9890	0.9890	0.9890
	S.D.	0.0232	0.0021	0.0014	2.28E – 16	2.28E – 16	2.28E – 16
<i>Iris</i>	Mean	0.7204	0.7774	0.8392	0.8832	0.8965	0.8992
	S.D.	0.1954	0.1300	0.0941	0.0806	0.0031	0.0073
<i>Newthyroid</i>	Mean	0.6555	0.5690	0.6001	0.6510	0.6564	0.6860
	S.D.	0.1462	0.1637	0.0877	0.1512	0.0528	0.0194
<i>Seeds</i>	Mean	0.8021	0.8573	0.8746	0.8949	0.8945	0.8954
	S.D.	0.1145	0.0944	0.0446	0.0066	0.0021	2.36E – 11
<i>Yeast</i>	Mean	0.2125	0.0956	0.1477	0.2227	0.2255	0.4247
	S.D.	0.0985	0.0710	0.0900	0.0910	0.1080	0.0241
<i>Glass</i>	Mean	0.2904	0.3263	0.3461	0.4036	0.4715	0.5518
	S.D.	0.1551	0.1004	0.1151	0.1165	0.0783	0.0039
<i>Wine</i>	Mean	0.7200	0.7216	0.7209	0.7250	0.7270	0.7273
	S.D.	0.0068	0.0044	0.0036	0.0047	0.0034	0.0034
<i>Lung Cancer</i>	Mean	0.4122	0.2471	0.2323	0.4178	0.5432	0.5790
	S.D.	0.1356	0.1273	0.1291	0.1329	0.0873	1.92E – 17

6. Conclusions

An extended membrane system combining an extended cell-like P system and the PSO mechanism, called the PSO-CP, is developed to solve clustering problems. This extended P system under the framework of membrane computing, using a cell-like P system with active membranes, integrates the rules for objects

and membranes. Different from the existing evolutionary clustering techniques, the PSO-CP uses basic evolution rules for objects based on the standard velocity and position updating rules of the particles in PSO and the communication rules for objects to transfer the best objects between membranes. Subsystems containing membranes are specially designed to avoid prematurity, and a modified evolution mechanism for objects is

used to evolve objects in the subsystems. The PSO-CP is evaluated on ten test datasets from the Artificial Datasets and the UCI Machine Learning Repository, and the computational results clearly exhibit the effectiveness of this proposed membrane system in solving clustering problems as compared with five existing PSO clustering methods.

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 membrane system uses the cell-like P system as the computation structure, and the communication rules between 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 the PSO-CP using large datasets. Balancing the local and global search abilities is also a hard problem to resolve in future studies. Future studies may also focus on extended membrane systems based on tissue-like P systems and other bio-inspired computing models. More studies are needed to apply these extended membrane systems to solve automatic and multiobjective clustering problems.

Data Availability

The seven artificial datasets 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 seven real-life datasets are from the UCI Machine Learning Repository, available at <http://archive.ics.uci.edu/ml/datasets.html> (accessed June 2019).

Conflicts of Interest

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

Acknowledgments

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

References

- [1] Y. Zhao, X. Liu, and X. Yan, "A grid-based chameleon algorithm based on the tissue-like P system with promoters and inhibitors," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 6, pp. 3652–3658, 2016.
- [2] S. Wang, C. Yu, D. Sun, and X. Wei, "Research on speed optimization strategy of hybrid electric vehicle queue based on particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8250480, 14 pages, 2018.
- [3] X. Y. Liu, H. Liu, H. C. Duan, and W. P. Wang, "Particle swarm optimization based on dynamic niche technology with applications to conceptual design," *Computer Science*, vol. 38, no. 10, pp. 668–676, 2016.
- [4] J. Hou, E. Xu, and W. X. Liu, "Density based cluster growing via dominant sets," *Neural Processing Letters*, vol. 48, no. 2, pp. 933–954, 2017.
- [5] I. Hassan, "I-k-means+: an iterative clustering algorithm based on an enhanced version of the k-means," *Pattern Recognition*, vol. 79, pp. 402–413, 2018.
- [6] Y. Z. Safaa, M. Dzulkifli, S. Tanzila, A. M. Hazim, and R. Amjad, "Content-based image retrieval using PSO and k-means clustering algorithm," *Arabian Journal of Geosciences*, vol. 8, no. 8, pp. 6211–6224, 2014.
- [7] Z. Dong, H. Jia, and M. Liu, "An adaptive multi-objective genetic algorithm with fuzzy c-means for automatic data clustering," *Applied Soft Computing*, vol. 2018, pp. 679–691, 2018.
- [8] H. B. Nguyen, B. Xue, and P. Andreae, "PSO with surrogate models for feature selection: static and dynamic clustering-based methods," *Memetic Computing*, vol. 10, no. 3, pp. 291–300, 2018.
- [9] Y.-H. Yang, X.-B. Xu, S.-B. He, J.-B. Wang, and Y.-H. Wen, "Cluster-based niching differential evolution algorithm for optimizing the stable structures of metallic clusters," *Computational Materials Science*, vol. 149, pp. 416–423, 2018.
- [10] F. Zabihi and B. Nasiri, "A novel history-driven artificial bee colony algorithm for data clustering," *Applied Soft Computing*, vol. 71, pp. 226–241, 2018.
- [11] G. Hojat, K. Kamran, H. M. Sadegh, and A. Aarabi, "An improved feature selection algorithm based on graph clustering and ant colony optimization," *Knowledge-Based Systems*, vol. 159, pp. 270–285, 2018.
- [12] N. Netjinda, T. Achalakul, and B. Sirinaovakul, "Particle swarm optimization inspired by starling flock behaviour," *Applied Soft Computing*, vol. 35, pp. 411–422, 2015.
- [13] W. Song, W. Ma, and Y. Qiao, "Particle swarm optimization algorithm with environmental factors for clustering analysis," *Soft Computing*, vol. 21, no. 2, pp. 283–293, 2017.
- [14] R. Liu, J. Li, J. Fan, C. Mu, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.
- [15] H. Lassad, B. Mohamed, and C. Abdelkader, "Improved adaptive particle swarm optimization for optimization functions and clustering fuzzy modeling system," *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, vol. 26, no. 5, pp. 717–739, 2018.
- [16] B. Asgarali and H. Abdolreza, "An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms," *Applied Soft Computing*, vol. 67, pp. 172–182, 2018.
- [17] R. Pereira de Gusmão and F. A. T. de Carvalho, "Clustering of multi-view relational data based on particle swarm

- optimization,” *Expert Systems with Applications*, vol. 123, pp. 34–53, 2019.
- [18] S. Manju and C. J. Kumar, “Sustainable automatic data clustering using hybrid PSO algorithm with mutation,” *Sustainable Computing Informatics and Systems*, vol. 23, pp. 144–157, 2019.
- [19] K.-W. Huang, Z.-X. Wu, H.-W. Peng, M.-C. Tsai, Y.-C. Hung, and Y.-C. Lu, “Memetic particle gravitation optimization algorithm for solving clustering problems,” *IEEE Access*, vol. 7, pp. 80950–80968, 2019.
- [20] W. Z. Zhang, G. Q. Li, W. W. Zhang, J. Liang, and G. G. Yen, “A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization,” *Swarm and Evolutionary Computation*, vol. 50, pp. 10569–10587, 2019.
- [21] A. A. Habibollah, J. S. Mohammad, and T. Mehdi, “A clustering algorithm based on integration of K -means and PSO,” in *Proceedings of the Swarm Intelligence and Evolutionary Computation*, pp. 59–63, Bam, Iran, March 2016.
- [22] G. Păun, “Computing with membranes,” *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.
- [23] Y. Z. Zhao, X. Y. Liu, and W. P. Wang, “Spiking neural P systems with neuron division and dissolution,” *Science China*, vol. 11, no. 9, Article ID e0162882, 2016.
- [24] A. Leporati, L. Manzoni, G. Mauri, A. E. Porreca, and C. Zandron, “The counting power of P systems with anti-matter,” *Theoretical Computer Science*, vol. 701, pp. 161–173, 2017.
- [25] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, “Dynamic threshold neural P systems,” *Knowledge-Based Systems*, vol. 163, pp. 875–884, 2019.
- [26] H. Peng, J. Yang, J. Wang et al., “Spiking neural P systems with multiple channels,” *Neural Networks*, vol. 95, pp. 66–71, 2017.
- [27] H. Peng and J. Wang, “Coupled neural P systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1672–1682, 2019.
- [28] J. Wang, H. Peng, W. Yu et al., “Interval-valued fuzzy spiking neural P systems for fault diagnosis of power transmission networks,” *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 102–109, 2019.
- [29] H. Peng, J. Wang, J. Ming et al., “Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems,” *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4777–4784, 2018.
- [30] J. Xue and X. Liu, “Lattice based communication P systems with applications in cluster analysis,” *Soft Computing*, vol. 18, no. 7, pp. 1425–1440, 2014.
- [31] X. 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.
- [32] X. Liu, Y. Zhao, and M. Sun, “An improved apriori algorithm based on an evolution-communication tissue-like P system with promoters and inhibitors,” *Discrete Dynamics in Nature and Society*, vol. 2017, no. 4, Article ID 6978146, 11 pages, 2017.
- [33] H. Peng, J. Wang, P. Shi, R. N. Agustín, and P. J. Marío, “An automatic clustering algorithm inspired by membrane computing,” *Pattern Recognition Letters*, vol. 68, pp. 34–40, 2015.
- [34] H. Peng, J. Wang, P. Shi, M. J. Pérez-Jiménez, and A. Riscos-Núñez, “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.
- [35] H. Peng, P. Shi, J. Wang, A. Riscos-Núñez, and M. J. Pérez-Jiménez, “Multiobjective fuzzy clustering approach based on tissue-like membrane systems,” *Knowledge-Based Systems*, vol. 125, pp. 74–82, 2017.
- [36] Y. H. Wang, X. Y. Liu, and L. S. Xiang, “GA-based membrane evolutionary algorithm for ensemble clustering,” *Computational Intelligence Neuroscience*, vol. 2017, no. 3, Article ID 4367342, 11 pages, 2017.
- [37] J. Xiao, Y. Huang, Z. Cheng, J. He, and Y. Niu, “A hybrid membrane evolutionary algorithm for solving constrained optimization problems,” *Optik*, vol. 125, no. 2, pp. 897–902, 2014.
- [38] J. Xiao, J.-J. He, P. Chen, and Y.-Y. Niu, “An improved dynamic membrane evolutionary algorithm for constrained engineering design problems,” *Natural Computing*, vol. 15, no. 4, pp. 579–589, 2016.
- [39] G. Singh and K. Deep, “Effectiveness of new multiple-PSO based membrane optimization algorithm on CEC 2014 benchmarks and iris classification,” *Natural Computing*, vol. 16, no. 3, pp. 473–496, 2017.
- [40] N. Elkhani, R. C. Muniyandi, and G. Zhang, “Multi-objective binary PSO with kernel P system on GPU,” *International Journal of Computers Communications and Control*, vol. 13, no. 3, pp. 323–336, 2018.
- [41] M. Clerc and J. Kennedy, “The particle swarm—explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [42] X. D. Li, “A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio,” in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation—GECCO ’07*, vol. 2007, pp. 78–85, ACM, London, UK, July 2007.
- [43] X. Y. Liu, L. S. Xiang, and X. Wang, “Spatial cluster analysis by the Adleman-lipton DNA computing model and flexible grids,” *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 894207, 32 pages, 2012.
- [44] X. Liu and A. Xue, “Communication P systems on simplicial complexes with applications in cluster Analysis,” *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 415242, 17 pages, 2012.
- [45] J. Xue, A. Camino, S. T. Bailey, X. Liu, D. Li, and Y. Jia, “Automatic quantification of choroidal neovascularization lesion area on OCT angiography based on density cell-like P systems with active membranes,” *Biomedical Optics Express*, vol. 9, no. 7, pp. 3208–3219, 2018.
- [46] X. Liu, Y. Zhao, and W. Sun, “Tissue P systems with cooperating rules,” *Chinese Journal of Electronics*, vol. 27, no. 2, pp. 324–333, 2018.
- [47] S. Rana, S. Jasola, and R. Kumar, “A review on particle swarm optimization algorithms and their applications to data clustering,” *Artificial Intelligence Review*, vol. 35, no. 3, pp. 211–222, 2011.
- [48] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 2002.
- [49] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks 1995*, pp. 1942–1948, Perth, Australia, November 1995.
- [50] B. Song, M. J. Pérez-Jiménez, and L. Pan, “Computational efficiency and universality of timed P systems with membrane

- creation,” *Soft Computing*, vol. 19, no. 11, pp. 3043–3053, 2015.
- [51] Artificial Datasets, <https://www.isical.ac.in/content/databases>.
- [52] H. C. Blake, *UCI Repository of Machine Learning Databases*, Academic Press, Cambridge, MA, USA, 1998.
- [53] K. Veeramachaneni, T. Peram, C. Mohan, and L. A. Osadciw, “Optimization using particle swarms with near neighbor interactions,” *Genetic and Evolutionary Computation—GECCO 2003*, vol. 2723, pp. 110–121, Springer, Berlin, Germany, 2003.
- [54] B. K. Kumari and S. P. Kumar, “Chaotic gradient artificial bee colony for text clustering,” *Soft Computing*, vol. 20, no. 3, pp. 1113–1126, 2016.
- [55] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 1–27, 1937.
- [56] H. Peng, J. Wang, P. J. Marío, and R. N. Agustín, “An unsupervised learning algorithm for membrane computing,” *Information Sciences*, vol. 304, pp. 80–91, 2015.