IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 1, 2020, accepted February 9, 2020, date of publication February 13, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973613

Stochastic Numerical P Systems With Application in Data Clustering Problems

JINYU YANG[1], HONG PENG[1], (Member, IEEE), XIAOHUI LUO[1], AND JUN WANG[2]

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[2]School of Electrical Engineering and Electronic Information, Xihua University, Chengdu 610039, China

Corresponding author: Hong Peng (ph.xhu@hotmail.com)

This work was supported in part by the Research Fund of Sichuan Science and Technology Project under Grant 2018JY0083, and in part by the Research Foundation of the Education Department of Sichuan Province, China, under Grant 17TD0034.

ABSTRACT This paper proposes an extension of numerical P (NP) systems, called stochastic numerical P (StNP) systems. In StNP systems, a novel stochastic production function-repartition protocol is developed to evolve the variables. Compared with usual production function-repartition protocol in NP systems, there are two differences in stochastic production function-repartition protocol: (i) some stochastic/randomized factors are introduced in production function and/or repartition protocol; (ii) data updating way is relaxed and can be flexibly used. An StNP system has a nested structure and contains some variables and a group of programs. The programs consist of stochastic production function-repartition protocols and/or usual production function-repartition protocols. Therefore, StNP systems are a class of distributed parallel computing models with stochastic and dynamic characteristics. Data clustering problems are used as an application to demonstrate the availability and effectiveness of StNP systems. Based on StNP systems, a novel partition clustering algorithm is presented. Experimental results demonstrate advantage of StNP systems for data clustering problems.

INDEX TERMS Membrane computing, numerical P systems, stochastic numerical P systems, stochastic production function-repartition protocol, data clustering.

I. INTRODUCTION

Membrane computing initiated by Gh. Păun [1], is a class of distributed parallel computing models, abstracted by the structure and functioning of living cells and the cooperation of cells in tissues, organs, and biological nervous systems [2]. Because of Păun's contribution, the class of models are called P systems in membrane computing domain. Motivated by different biological mechanisms, a variety of P systems have been investigated in the past two decades [3]–[16]. There are three main types in P systems: cell-like P systems, tissue-like P systems and neural-like P systems, which adopt a nested, a network-like and a directed graph structures, respectively. In addition to membrane structure, a P system contains several important components: data (objects), rules and methods to control the rules. The data is used to express the states of cells (or neurons), while rules describe the dynamic behavior of the system. It has been proven that most of P systems were Turing-universal and able to solve some NP hard problems in a feasible time. Moreover, these models have been used to tackle real application problems [17], for example, knowledge representation [18], [19], numerical optimization [20], [21], image and signal processing [22]–[26], fault diagnosis [27]–[29], ecology and system biology [30]–[32].

Numerical P (NP) systems [33] are a variant of cell-like P systems, motivated by the nested structure of cells and the idea of distribution profits in economics. Usually, NP systems use a nested topology, however, numeric-type variables instead of multiset of string in cell-like P systems are used and the behavior of the system is controlled by a group of programs, which are called production function-repartition protocols. A production function-repartition protocol consists of two parts: (i) production function; (ii) repartition protocol. The first part uses local variables to compute a function value. The computed value is allocated to the variables from the region where the program resides, and to the variables in its upper and lower neighboring regions based on the repartition protocol. The values of these variables are evolved by means of programs. Therefore, NP systems are a kind of distributed parallel computing models with dynamic

The associate editor coordinating the review of this manuscript and approving it for publication was Shagufta Henna.

characteristic. However, NP systems are the deterministic systems.

In a lot of real applications, the methods to solve them may involve some stochastic/randomized mechanisms. However, usual production function-repartition protocol in NP systems lacks the stochastic/randomized mechanisms. Hence, NP systems may fail to deal with these applications. To overcome the limitation, a novel extension of production function-repartition protocol that contains some stochastic/randomized factors is developed, called stochastic production function-repartition protocol. Based on the extension, a novel variant of NP systems, stochastic numerical P (StNP) systems, is proposed in this work. Different from NP systems, StNP systems have two features:

(1) Since production function and/or repartition protocol contain some stochastic/randomized factors, StNP systems can exhibit stochastic characteristic.
(2) the data updating way is relaxed and can be flexibly used in StNP systems.

Therefore, StNP systems are a class of distributed parallel computing models with stochastic and dynamic characteristics. To evaluate the availability and effectiveness of StNP systems, data clustering problems are used as an application. For this purpose, an StNP system is considered, where a specific stochastic production function-repartition protocol is designed. Based on the StNP system, we develop a novel partition clustering algorithm that can search for the optimal cluster centers by using the stochastic production function-repartition protocol.

The remainder of this paper is organized as follows. Section II briefly reviews the related work of NP systems and data clustering problems. Section III describes the proposed StNP systems in detail. Section IV discusses the application of StNP systems, and an StNP systems-based partition clustering algorithm is developed. Conclusions and future work is given in Section V.

II. RELATED WORK

A. NUMERICAL P SYSTEMS

Numerical P (NP) systems proposed by Păun and Păun [33] is a variant of cell-like P systems. NP systems has a nested topology and numerical variables instead of multiset of string, while the behavior of the systems is characterized by a group of programs.

Definition 1: An NP system [33] is defined by a construct

$$\Pi = (m, H, \mu, (V_1, P_1, V_1(0)), \ldots, (V_m, P_m, V_m(0)), V_{in}, V_{out}) \quad (1)$$

where

(1) $m \geq 1$ is the number of membranes;
(2) H is an alphabet of labels for membranes in Π;
(3) μ is a rooted tree with m nodes labeled by the symbols of H;
(4) V_i is the set of variables in region i, $1 \leq i \leq m$;
(5) $V_i(0)$ is the set of initial values of the variables in region i, $1 \leq i \leq m$;
(6) P_i is the set of programs in region i, $1 \leq i \leq m$; every program has of the form:

$$F_{l,i}(x_{1,i}, \ldots, x_{k_i,i}) \rightarrow c_{l,i,1}|v_{l,i,1} + \cdots + c_{l,i,l_i}|v_{l,i,l_i} \quad (2)$$

where $F_{l,i}(x_{1,i}, \ldots, x_{k_i,i})$ is the production function of the program, and $RP_{l,i} = c_{l,i,1}|v_{l,i,1} + \cdots + c_{l,i,l_i}|v_{l,i,l_i}$ is the corresponding repartition protocol;
(7) V_{in} and V_{out}, respectively, denote the sets of input variables and output variables.

The variables in the system will be evolved by programs during the computation. Each program is a production function-repartition protocol. Therefore, the variables describe the state of the system, while the programs consisting of production function-repartition protocols characterize its dynamic behavior.

The variables are updated by production function-repartition protocols. Denote by $C_{l,i} = \sum_{s=1}^{l_i} c_{l,i,s}$ the sum of all coefficients of the repartition protocol, and the "unitary portion" is computed by $q_{l,i}(t) = F_{l,i}(x_{1,i}, \ldots, x_{k_i,i})/C_{l,i}$. Thus, the value that is added to variable $v_{l,i,r}$ is $\triangle_{l,i,r}(t) = q_{l,i}(t) \times c_{l,i,r}$. If variable $v_{l,i,r}$ exists in some applied programs, for instance, $RP_{l_1,i_1}, \ldots, RP_{l_k,i_k}$, then all these values $\triangle_{l_1,i_1,r}, \ldots, \triangle_{l_k,i_k,r}$ are added to variable $v_{l,i,r}$.

In NP systems, there are two data updating ways:

(1) Substitution updating way:
If at time t, variable $v_{l,i,r}$ appears in at least a production function of the applied programs and exists in some repartition protocols $RP_{l_1,i_1}, \ldots, RP_{l_k,i_k}$ of the applied programs, then at time $t+1$, its value is computed by $v_{l,i,r}(t+1) = \sum_{s=1}^{k} \triangle_{l_s,i_s,r}(t)$.
(2) Increment updating way:
If at time t, variable $v_{l,i,r}$ appears only in some repartition protocols $RP_{l_1,i_1}, \ldots, RP_{l_k,i_k}$ of the applied programs but not in any production function of the applied programs, then at time $t+1$, its value is computed by $v_{l,i,r}(t+1) = v_{l,i,r}(t) + \sum_{s=1}^{k} \triangle_{l_s,i_s,r}(t)$.

Remark 1: In NP systems and variants, substitution updating way is used in the case that variable $v_{l,i,r}$ exists in production functions, and in other cases, increment updating way is used. In the proposed StNP systems, however, the requirement is relaxed, and the two data updating ways can be used flexibly.

In recent years, several variants of NP systems have been proposed. By introducing enzymatic variables, Pavel *et al.* [34] proposed enzymatic numerical P systems, where whether a program is executed depends on the value of the corresponding enzymatic variable. Zhang and Pan [35] investigated numerical P systems with thresholds. Similar to the enzymatic variable, the threshold was regarded as a control mechanism: when each variable in a production function is higher (resp., lower) than a threshold, the program can be executed. The two cases correspond to lower-threshold and upper-threshold, respectively. Pan *et al.* [36] investigated numerical P systems with production thresholds, where threshold control mechanism is reflected on the function value of production function. Liu *et al.* [37] presented a more general control mechanism, i.e., Boolean condition. Usually, the variables are located at different regions, and their positions are given initially and remain unchanged during the computation. The communicating objects through membranes is one of important features of P systems. According to the mechanism, Zhang *et al.* [38] discussed numerical P systems with migrating variables. Moreover, simulation softwares of NP systems and their applications have been investigated, for example, robot [39]–[44]. In addition, computing power of NP systems and variants has been discussed in the recent years [45]–[49].

B. DATA CLUSTERING PROBLEMS

Data clustering can be viewed as such a task that partitions q data samples into several groups based on some similarity measure given in advance. It is essentially a NP-hard problem to minimize some similarity metric for finding the partitioning on a nonuniform dataset [50], [51].

Suppose that $D = \{X_1, X_2, \ldots, X_q\} \subseteq R^{q\times d}$ is a data set consisting of q unlabeled data samples in R^d, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, and x_{ij} denotes the jth real value feature of the ith sample. The objective of data clustering method is to search for an optimal partitioning of the data set, which partitions D to K clusters, $D_1, D_2, \ldots, D_K$, where the similarity of data samples in the same cluster is very high, and simultaneously the similarity of data samples in different clusters is very low. Usually, the partitioning holds the following properties:

(1) Each cluster contains at least a data sample, i.e., $D_i \neq \phi$.

(2) Two different clusters contain no data sample in common, i.e., $D_i \cap D_j = \phi$, $\forall i \neq j$ and $i, j = 1, 2, \ldots, K$.

(3) Each data sample is definitely assigned to a cluster, i.e., $\cup_{i=1}^{K} D_i = D$.

K-means is a popular clustering algorithm, which determines the optimal cluster centers for a data set. Usually, to find the optimal cluster centers, data clustering is regarded as an optimization problem, whose objective function can be expressed by

$$J_m(z_1, z_2, \ldots, z_K) = \sum_{k=1}^{K} \sum_{X_j \in D_k} ||X_j - z_k||^2 \tag{3}$$

where $z_1, z_2, \ldots, z_K$ are cluster centers of $D_1, D_2, \ldots, D_K$. Consequently, the data clustering problem is formalized as follows:

$$\min_{z_1, z_2, \ldots, z_K} \left[J_m(z_1, z_2, \ldots, z_K) = \sum_{k=1}^{K} \sum_{X_j \in D_k} ||X_j - z_k||^2 \right], \tag{4}$$

where $z_1, z_2, \ldots, z_K$ denote K parameters that correspond to a feasible set of cluster centers.

To find the optimal cluster centers, k-means applies the average of samples in each cluster to update the its cluster center constantly. However, there are several shortcomings in k-means algorithm: (i) it is easy to fall into local minima, (ii) it is heavily dependent on the initial cluster centers, and (iii) it is time consuming for finding the global optimal centers when the size of data is large. To address these shortcomings, seveal evolutionary clustering methods were investigated according to genetic algorithms (GA) [52], particle swarm optimization (PSO) [53], differential evolution (DE) [54] and artificial bee colony (ABC) [55].

In the last years, application of P systems in data clustering problems has received a lot of attention. Xue and Liu [56] proposed a clustering algorithm where lattice-based communication P systems were applied to handle data clustering. In Peng et al. [57], an evolution-communication P system was applied to develop a fuzzy clustering method. A P systems-based hybrid clustering algorithm was investigated [58], which used hybrid evolution mechanism to evolve the set of cluster centers. Two automatic membrane clustering algorithms have been investigated [59], [60], where a representation with control bits and a membrane system with active membranes were applied to implement two automatic clustering mechanisms. Peng *et al.* [61] discussed a multi-objective fuzzy clustering method using tissue-like P systems. A cluster splitting method by P systems and Hopfield networks was investigated in Liu *et al.* [62]. In addition, Yan *et al.* [63] discussed a hybrid chain-hypergraph P system for multiobjective ensemble clustering.

III. STOCHASTIC NUMERICAL P SYSTEMS

In this section, we first introduce stochastic production function-repartition protocol, and then present stochastic numerical P (StNP) systems.

A. STOCHASTIC PRODUCTION FUNCTION-REPARTITION PROTOCOL

By introducing some stochastic/randomized factors, an extension of production function-repartition protocol is developed, called stochastic production function-repartition protocol. The stochastic production function-repartition protocol can be described by

$$\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i}) \rightarrow \tilde{c}_{1,l,i}|x_{1,i} + \cdots + \tilde{c}_{k_i,l,i}|x_{k_i,i} \tag{5}$$

where $\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i})$ is stochastic production function, and $\tilde{c}_{1,l,i}|x_{1,i} + \cdots + \tilde{c}_{k_i,l,i}|x_{k_i,i}$ is stochastic repartition protocol. Note that $v_{l,i,1}, \ldots, v_{l,i,l_i}$ are the variables in region i and/or the variables in the corresponding upper (parent) and lower (children) regions, while $x_{1,i}, \ldots, x_{k_i,i}$ are only the variables in region i.

Since $\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i})$ generally contains some stochastic/randomized factors, it is called stochastic production function. Of course, $\tilde{F}_{l,i}$ can also be degenerated as usual production function $F_{l,i}$ in NP systems when it contains no random factors. The $\tilde{c}_{1,l,i}|x_{1,i} + \cdots + \tilde{c}_{k_i,l,i}|x_{k_i,i}$ is stochastic repartition protocol, where $\tilde{c}_{1,l,i}, \ldots, \tilde{c}_{k_i,l,i}$ are k_i random real numbers in [0, 1]. Similarly, denote by $\tilde{C}_{l,i} = \sum_{s=1}^{k_i} \tilde{c}_{s,l,i}$ the sum of all coefficients of the stochastic repartition protocol at time t, and the "unitary portion" is computed by $\tilde{q}_{l,i}(t) = \tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i})/\tilde{C}_{l,i}$. Thus, the value that is added to variable $x_{r,i}$ is $\tilde{\Delta}_{r,l,i}(t) = \tilde{q}_{l,i}(t) \times \tilde{c}_{r,l,i}$. If variable $x_{r,i}$ exists in several applied programs, for instance, $RP_{l_1,i_1}, \ldots, RP_{l_k,i_k}$, then all these values $\tilde{\Delta}_{r,l_1,i_1}, \ldots, \tilde{\Delta}_{r,l_k,i_k}$ are added to variable $x_{r,i}$.

As mentioned above, two data updating ways can be flexibly used in stochastic production function-repartition protocol, especially when dealing with real applicatons. That is, one of the two data updating ways can be used whether a variable is involved the stochastic production function or in the stochastic repartition protocol. Therefore, we have

(1) If substitution updating way is used, then at time $t+1$, new state of variable $x_{r,i}$ is computed by

$$x_{r,i}(t+1) = \sum_{s=1}^{k} \tilde{\Delta}_{r,l_s,i_s}(t);$$

(2) If increment updating way is used, then at time $t+1$, new state of variable $x_{r,i}$ is computed by

$$x_{r,i}(t+1) = x_{r,i}(t) + \sum_{s=1}^{k} \tilde{\Delta}_{r,l_s,i_s}(t).$$

In summary, different from usual production function-repartition protocol, the production function and/or repartition protocol in stochastic production function-repartition protocol involve in some stochastic/randomized factors, and two data updating ways can be flexibly used.

B. STOCHASTIC NUMERICAL P SYSTEMS

Definition 2: An StNP system is defined by a construct

$$\Pi = (m, H, \mu, (V_1, P_1, V_1(0)), \ldots, (V_m, P_m, V_m(0)), V_{in}, V_{out}) \tag{6}$$

where

(1) $m \geq 1$ is the number of membranes;

(2) H is an alphabet of labels for membranes in Π;

(3) μ is a rooted tree with m nodes labeled with the symbols of H;

(4) V_i is the set of variables in region i, $1 \leq i \leq m$;

(5) $V_i(0)$ is the set of initial values of the variables in region i, $1 \leq i \leq m$;

(6) P_i is the set of programs in region i, $1 \leq i \leq m$; each program has the following form:

$$\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i}) \rightarrow \tilde{c}_{1,l,i}|x_{1,i} + \cdots + \tilde{c}_{k_i,l,i}|x_{k_i,i} \tag{7}$$

where $\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i})$ is stochastic production function, and $\tilde{c}_{1,l,i}|x_{1,i} + \cdots + \tilde{c}_{k_i,l,i}|x_{k_i,i}$ is stochastic repartition protocol of the program;

(9) V_{in} and V_{out}, respectively, denote the sets of input variables and output variables.

The variables in the system will be evolved by programs during the computation. Each program consists of two parts: stochastic production function and stochastic repartition protocol. Therefore, the execution of each program includes two phases: production phase and repartition phase.

(1) Production phase. According to the values of its variables, production function $\tilde{F}_{l,i}(v_{l,i,1}, \ldots, v_{l,i,l_i})$ is computed. Note that $v_{l,i,1}, \ldots, v_{l,i,l_i}$ are the variables in region i and/or the variables in the corresponding upper (parent) and lower (children) regions.

(2) Distribution phase: the computed value of production function is allocated to the variables from the region where the program resides, i.e., $x_{1,i}, \ldots, x_{k_i,i}$.

Notice that the two phases take place in one time unit. As mentioned above, two data updating ways can be flexibly used in StNP systems. If substitution updating way is used in a stochastic production function-repartition protocol, then once the production phase is completed, the variables contained in the stochastic production function are reset to zero; otherwise, increment updating way is applied, that is, the variables cannot be reset to zero.

In general, StNP systems can work in one of the following modes:

(1) all-parallel mode: at each time unit, in each membrane, all enabled programs are applied, allowing that more than one program share the same variable;

(2) one-parallel mode: apply programs in the all-parallel mode with the restriction that a variable can only exist in one of the applied programs; in the case of multiple choices, the programs to be applied are chosen in the non-deterministic way;

(3) sequential mode: at each time unit, only a program is applied in each membrane; if more than one program in a membrane can be applied, then one of them is chosen non-deterministically.

A configuration is defined by a real vector consisting of the values of all variables of an StNP system at a computing step. Initially, the variables are specified by the values in $V_i(0)$, $1 \leq i \leq m$. Applying the programs in the working way above, a transition of the system from a configuration to the next one can be defined. A computation is defined as a sequence of such transitions. If no program in each region can be applied, then a halting configuration attains.

IV. APPLICATION IN DATA CLUSTERING PROBLEMS

A. A CLUSTERING ALGORITHM BASED ON STNP SYSTEMS

Based on StNP systems, a novel clustering algorithm is presented to handle data clustering problems. The core of the algorithm is an StNP system of degree $(2m+1)$, Π_1, which is described as follows.

$$\begin{aligned}\Pi_1 = (&2m+1, H, \mu, (V_1, P_1, V_1(0)), \ldots, (V_m, P_m, V_m(0)),\\ &(V_1', P_1', V_1'(0)), \ldots, (V_m', P_m', V_m'(0)),\\ &(V_0'', P_0'', V_0''(0)), V_{out})\end{aligned} \tag{8}$$

where

(1) $2m+1$ is the number of membranes;

(2) $H = \{0, 1, 2, \ldots, m, 1', 2', \ldots, m'\}$ is an alphabet of labels for membranes in Π_1;

(3) μ denotes a rooted tree with $2m+1$ nodes labeled with the elements of H;
(4) V_i denotes the set of variables in region i, $1 \le i \le m$; V'_i denotes the set of variables in region i', $1' \le i' \le m'$; V''_0 denotes the set of variables in region 0;
(5) $V_i(0)$ is the set of initial values of the variables in region i, $1 \le i \le m$; $V'_i(0)$ is the set of initial values of the variables in region i', $1' \le i' \le m'$; $V''_0(0)$ is the set of initial values of the variables in region 0;
(6) P_i is the set of programs in region i, $1 \le i \le m$; P'_i is the set of programs in region i', $1' \le i' \le m'$; P''_0 is the set of programs in region 0;
(7) V_{out} is the set of output variables.

In the following, several important components of system Π_1 are explained/illustrated in detail.

1) THE NESTED MEMBRANE STRUCTURE

The designed StNP system Π_1 has $2m+1$ membranes, which form a nested membrane structure, shown in Figure 1. The $2m+1$ membranes are labeled by $0, 1, \ldots, m, 1', \ldots, m'$, respectively. The regions that correspond to the $2m+1$ membranes are denoted by $R''_0, R_1, \ldots, R_m, R'_1, \ldots, R'_m$, respectively. From the perspective of the functioning, region R_i is used to evolve the set of cluster centers, called evolution region, $1 \le i \le m$. Regions $R'_1, \ldots, R'_m, R''_0$ are called storage regions, where R'_i stores the best value of variables found in region R_i so far, $1 \le i \le m$, while R''_0 is used to remember the best value of variables found in the system so far.

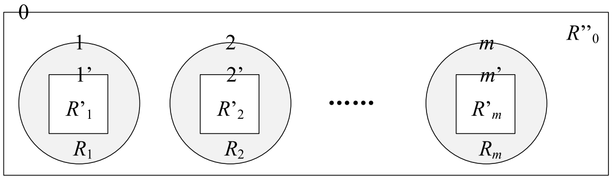


FIGURE 1. The nested structure used in StNP system Π_1.

2) VARIABLES

Due to different roles of the regions, the variables in the system Π_1 can be divided as three categories and they are located at different regions. The three kinds of variables are illustrated as follows.

(1) Variables in V_i, $1 \le i \le m$.
The goal of designing the StNP system Π_1 is to determine the optimal set of cluster centers for dataset D. Therefore, each variable in V_i denotes a candidate set of cluster centers. For example, $z_1, z_2, \ldots, z_K$ are a group of cluster centers, where $z_k = (z_{k1}, z_{k2}, \ldots, z_{kd})$ is kth cluster center. A variable Z in V_i denotes a set of cluster centers, so formally $Z = (z_1, z_2, \ldots, z_K) = (z_{11}, z_{12}, \ldots, z_{1d}, \ldots, z_{K1}, z_{K2}, \ldots, z_{Kd}) \in R^{K\times d}$.
For simply, assume that each V_i has the same number of variables, n, i.e., $|V_i| = n$, $1 \le i \le m$. Denote by $Z^i_j \in R^{K\times d}$ the jth variable in V_i, thus, $V_i = \{Z^i_1, Z^i_2, \ldots, Z^i_n\}$, indicating n groups of candidate cluster centers in region i.
(2) Variables in V'_i, $1 \le i \le m$.
There is only one variable in V'_i, denoted by Z^i_{lbest}, which is used to store the best value with best fitness found in V_i so far.
(3) Variables in V''_0.
There is only one variable in V''_0, expressed by Z_{gbest}, which is used to store the best value with best fitness found in the system so far.

3) INITIAL VALUES OF VARIABLES

Initially, the initial variable values in StNP system Π_1 are assigned by an initialization program, including variable values in $V_i(0)$, $V'_i(0)$ and $V''_0(0)$.

(1) Initial variable values in $V_i(0)$, $1 \le i \le m$.
The initialization program assigns a value for each variable in V_i by generating a group of clustering centers randomly. Assume that A_l and B_l are the lower and upper bounds of samples in data set D in lth dimension respectively, $1 \le l \le d$. For each variable Z^i_j in V_i, its initial value in $V_i(0)$ can be generated by an initialization algorithm (see Algorithm 1). At the same time, the initialization algorithm computes the fitness value of each variable Z^i_j in V_i according to the following initial value.
(2) Initial variable values in $V'_i(0)$, $1 \le i \le m$.
Z^i_{lbest} is the only variable in V'_i. Suppose Z^i_s is the variable with minimum fitness value in V_i, i.e., $f(Z^i_s(0)) = \min\limits_{1\le j\le n} f(Z^i_j(0))$. Therefore, initially $Z^i_{lbest} = Z^i_s$ and $Z^i_{lbest}(0) = Z^i_s(0)$.
(3) Initial variable values in V''_0.
Z_{gbest} is only one variable in V''_0. Suppose Z^t_s is the variable with minimum fitness value in all $V'_i(1 \le i \le m)$, i.e., $f(Z^t_s(0)) = \min\limits_{\substack{1\le i\le m\\ 1\le j\le n}} f(Z^i_j(0))$. Therefore, initially $Z_{gbest} = Z^t_s$ and $Z_{gbest}(0) = Z^t_s(0)$.

Note that in Algorithm 1, *random*(A_l, B_l) is a random function that generates a random real number in $[A_l, B_l]$.

Algorithm 1 The Initialization Algorithm

```
for $i = 1$ to $m$
  for $j = 1$ to $n$
    for $k = 1$ to $K$
      for $l = 1$ to $d$
        $z_{i,j,k,l}(0) = random(A_l, B_l)$;
      end for
      $z_{i,j,k}(0) = (z_{i,j,k,1}(0), \ldots, z_{i,j,k,d}(0))$;
    end for
    $Z^i_j(0) = (z_{i,j,1}(0), \ldots, z_{i,j,K}(0))$;
  end for
end for
```

4) FITNESS FUNCTION

In the presented clustering algorithm, a fitness function is introduced to evaluate the variables in the system Π_1. The fitness function is defined by

$$f(Z_j^i) = f(z_{i,j,1}, \ldots, z_{i,j,K}) = \sum_{k=1}^{K} \sum_{X_l \in D_k} ||X_l - z_{l,i,k}||^2 \quad (9)$$

where $Z_j^i = (z_{i,j,1}, \ldots, z_{i,j,K})$ is the jth variable in V_i. Generally, the smaller the function value $f(Z_j^i)$, the better the clustering solution Z_j^i.

5) PROGRAMS

(1) Programs in P_i, $1 \leq i \leq m$.

P_i denotes the programs in region i, and the variables in V_i are evolved by programs in P_i. The programs in P_i are described as follows.

$$\begin{aligned}\tilde{F}_j^i &= \tilde{r}_1(Z_{lbest}^i - Z_j^i) + \tilde{r}_2(Z_{gbest} - Z_j^i)\\ &\rightarrow \tilde{c}_1|Z_1^i + \cdots + \tilde{c}_n|Z_n^i \end{aligned} \quad (10)$$

where $1 \leq i \leq m$ and $1 \leq j \leq n$; Z_j^i is the jth variable in V_i, Z_{lbest}^i is the variable in V_i', and Z_{lbest} is the variable in V_0''; $\tilde{r}_1, \tilde{r}_2 \in [0, 1]$ is two random number; $\tilde{c}_i (1 \leq i \leq n)$ are random numbers in [0, 1], and $\tilde{c}_1 + \ldots + \tilde{c}_n = 1$. Note that the programs in Eq.(10) use increment updating way to evolve the variables in V_i.

(2) Programs in P_i', $1 \leq i \leq m$.

The programs in P_i' are used to update the variable Z_{lbest}^i in V_i'. The programs are expressed as follows.

$$\tilde{F}'_i = Z_{j0}^i \text{ with } f(Z_{j0}^i) = \min_{1 \leq j \leq n} \{f(Z_j^i)\} \rightarrow 1|Z_{lbest}^i \quad (11)$$

where Z_{j0}^i is the variable with the smallest fitness value in V_i. The variable with the smallest fitness value is used to update the Z_{gbest}^i, and substitution updating way is used.

(3) Programs in P_0''.

The programs in P_0'' are used to update the variable Z_{gbest} in V_0''. The programs are expressed as follows.

$$\tilde{F''}_0 = Z_{j0}^{i0} \text{ with } f(Z_{j0}^{i0}) = \min_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \{f(Z_j^i)\} \rightarrow 1|Z_{gbest} \quad (12)$$

where Z_{j0}^{i0} is the variable with the smallest fitness value in all $V_i (1 \leq i \leq m)$. The variable with the smallest fitness value is used to update the Z_{gbest} according to substitution updating way.

6) HALTING AND OUTPUT

In the proposed clustering algorithm, a simple halting condition is used: maximum number of iterations. Starting with initial values in $V_i(0)(1 \leq i \leq m)$, $V_i'(0)(1 \leq i \leq m)$ and $V_0''(0)$, StNP system continues to execute the programs in $P_i(1 \leq i \leq m)$, $P_i'(1 \leq i \leq m)$ and P_0'' in parallel until it reaches the halting condition. At this time, the system halts and the value of variable Z_{gbest} is the output of the system. Suppose that $Z_{gbest} = (z_1, z_2, \ldots, z_K)$. Therefore, $z_1, z_2, \ldots, z_K$ are the optimal cluster centers for dataset D.

7) IMPLEMENTATION OF ALGORITHM

The core of the proposed clustering algorithm is an StNP system consisting of $2m + 1$ regions (membranes). The StNP system is applied to find the optimal set of cluster centers for a dataset. Figure 2 shows flow chart of the proposed clustering algorithm. According to the description of components of StNP system above, the proposed clustering algorithm can be briefly illustrated as follows.

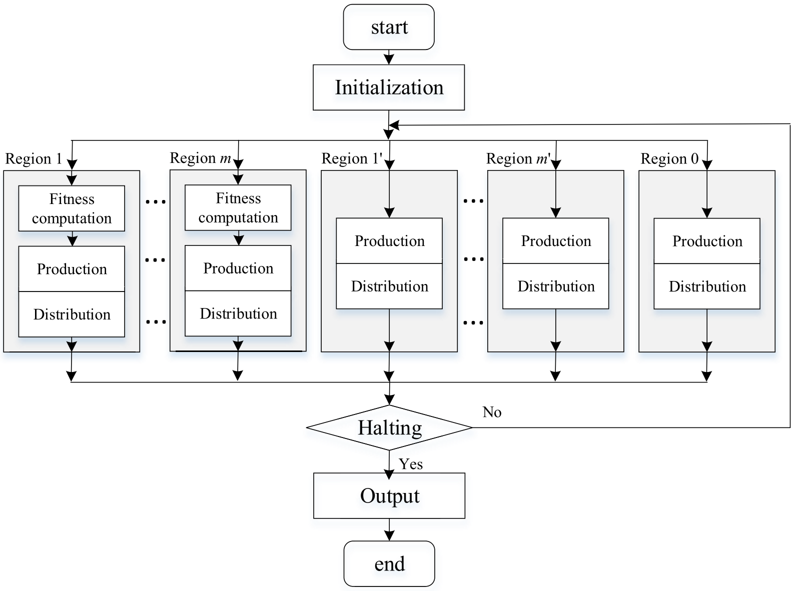


FIGURE 2. **The flow chart of the proposed clustering algorithm.**

The StNP system has $2m+1$ regions: (i) m evolution regions (from region 1 to region m), used to evolve the variables in the system; (ii) m storage regions (from region $1'$ to region m'); (iii) a global storage region 0. Each evolution region contains n variables (Z_j^i), each of which is applied to denote a set of candidate cluster centers; each storage region has only a variable (Z_{lbest}^i), which is used to store the best value of variable found in the following evolution region; the global storage region contains a variable (Z_{gbest}), which is used to remember the best value of variables found in the system.

The proposed clustering algorithm takes the data set D to be clustered as its input. Moreover, it has four priori parameters: (i) K (the number of clusters); (ii) m (the number of evolution regions); (iii) n (the number of variables in each evolution region); (iv) maximum number of iterations.

Initially, StNP system uses the initialization program (Algorithm 1) to randomly generate the initial values in $V_i(0)$, $V_i'(0)$ and $V_0''(0)$. The StNP system starts with the initial values. During the computation, all regions work in parallel, and the programs in each region are executed in all-parallel mode. StNP system runs continuously until halting condition attains. When it halts, the output is the value of variable (Z_{gbest}) in region 0, i.e., a group of optimal cluster centers.

B. SIMULATION EXPERIMENTS

This paper discusses a new variant of NP systems, called StNP systems, and proposes a novel clustering algorithm based on StNP systems. In this work, data clustering is used as an application of StNP systems. Therefore, the experiments are designed for evaluating the proposed clustering algorithm based on StNP systems.

1) DATA SETS

In experiments, ten benchmark datasets were used to evaluate the proposed clustering algorithm, including four two-dimensional artificial data sets (i.e., "AD_5_2", "Square_4", "Sym_3_22" and "Data_9_2", shown in Figure 4(a) and six real data sets from UCI [64] (i.e., "Iris", "Wine", "Livedisorder", "LungCancer", "Newthyroid" and "BreastCancer"). The data sets are described in Table 1. A clustering performance metric, J_m in Eq. (3), was adopted to evaluate the clustering performance of the proposed and the compared clustering algorithms. Generally, for each dataset, the smaller the J_m value, the better the corresponding algorithm.

2) THE COMPARED METHODS AND PARAMETER SET

To evaluate the clustering performance of the proposed clustering algorithm, it was compared with five state-of-the-art clustering algorithms that used the different optimization techniques. In experiments, the proposed clustering algorithm is denoted by NMCA, indicating a membrane clustering algorithm based on NP systems. The five compared algorithms as well as their parameters are illustrated as follows.

(1) MCA: a clustering algorithm inspired from membrane computing [65], where a tissue-like P system is used and three genetic operations (selection, crossover and mutation) are considered as evolution rules. The parameters are given: the system has 16 cells, every cell contains 100 objects and the number of iterations is 200.

(2) VSSO: an accelerated simplified swarm optimization [66]. In VSSO, parameters are set to $c_w = 0.2$, $c_g = 0.65$, $v = 0.2$, $d = 10$, $\beta = 0.1$, population size = 100 and the number of iterations = 200.

(3) BH: a black hole-based clustering algorithm [67]. The parameters are set as follows: the number of particles is 100, and the number of iterations is 200.

(4) KSRPSO: particle swarm optimization with selective particle regeneration for data clustering [68]. In KSRPSO, parameters are set to $c_1 = 0.5$, $c_2 = 2.5$, $f = 0.2$, $d = 0.2$, $a = 0.8$, population size = 100 and the number of iterations = 200.

(5) classical k-means algorithm [69]. Only the number of clusters is set for each data set according to Table 1.

TABLE 1. The description of data sets.

Data set	Number of data points (n)	Data dimension (d)	Number of Clusters (k)
AD_5_2	250	2	5
Square_4	1000	2	4
Sym_3_22	600	2	3
Data_9_2	900	2	9
Iris	150	4	3
Wine	178	13	3
Livedisorder	345	6	2
LungCancer	32	56	3
Newthyroid	215	5	3
BreastCancer	683	9	2

3) EFFECTS OF DIFFERENT PARAMETERS ON EXPERIMENTAL RESULTS

Usually, different parameters have different effects on the experimental results. Three priori parameters are analyzed as follows, including number of variables in cells, number of iterations and number of cells.

(1) number of variables and number of iterations.

Suppose that each region in StNP system has the same number of variables. Generally, the more the number of variables, the stronger the local exploitation capability, however, too many numbers of variables will increase the runtime of the algorithm. On the other hand, the number of iterations also has a certain influence on both clustering performance and runtime. Usually, a greater number of iterations enable it to have more opportunities to exploit the optimal solution, but it needs more running time.

Figure 3 gives average convergence curves of the proposed clustering algorithm with different parameters on

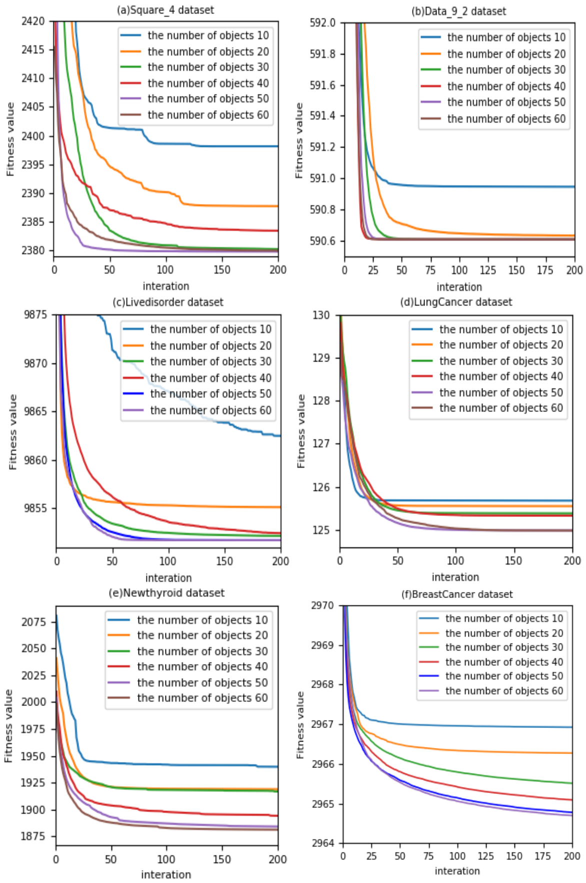


FIGURE 3. The clustering results of the proposed clustering algorithm using different parameters on six data sets.

TABLE 2. Effects of different number of cells for the proposed clustering algorithm on experimental results.

Data set	4 cells	8 cells	16 cells	20 cells
AD_5_2	326.588 (±0.173)	326.493 (±0.110)	326.446 (±0.029)	**326.439** (±**0.001**)
Square_4	2379.812 (±0.105)	2379.753 (±0.027)	2379.763 (±0.056)	**2379.745** (±**0.022**)
Sym_3_22	1247.695 (±0.002)	**1247.693** (±0.001)	**1247.693** (±**1.8e-06**)	**1247.693** (±0.0002)
Data_9_2	**590.606** (±**8.11e-06**)	**590.606** (±1.0e-05)	**590.606** (±1.0e-05)	**590.606** (±1.0e-05)
Iris	96.550 (±0.151)	96.502 (±0.134)	96.423 (±0.065)	**96.407** (±**0.028**)
Wine	16292.973 (±0.483)	16292.643 (±0.318)	16292.365 (±0.090)	**16292.288** (±**0.047**)
Livedisorder	9851.749 (±0.079)	9852.026 (±0.541)	**9851.765** (±0.108)	9851.838 (±**0.082**)
LungCancer	125.258 (±0.298)	125.254 (±0.363)	124.811 (±**0.123**)	**124.696** (±0.134)
Newthyroid	1876.982 (±7.184)	1866.899 (±0.682)	1867.456 (±1.070)	**1866.468** (±**0.003**)
BreastCancer	2964.630 (±0.385)	2964.486 (±0.209)	2964.531 (±0.275)	**2964.448** (±**0.132**)

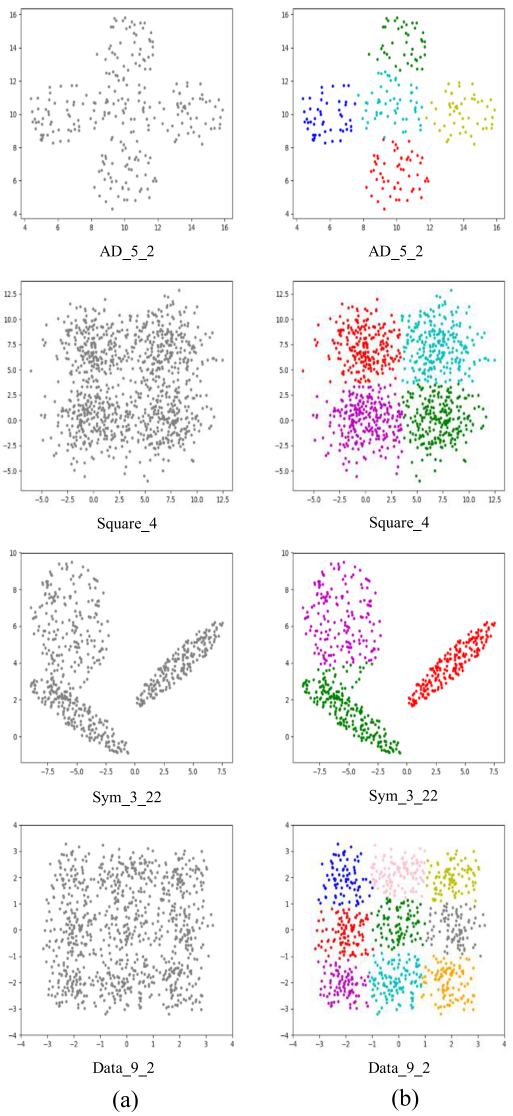


FIGURE 4. Artificial data sets and clustering results of the proposed clustering algorithm: (a) artificial data sets, and (b)clustering results.

six data sets, including Square_4, Data_9_2, Livedisorder, LungCancer, Newthyroid and BreastCancer, where the number of iterations is set to 200, and the number of variables is considered as 10, 20, 30, 40, 50 or 60 respectively. The results indicate that the proposed clustering algorithm achieves the best convergence performance in case of 50 variables in each region. Simultaneously, the proposed clustering algorithm is enough to complete the convergence within the 100 iterations.

(2) number of regions.
To find the effect of different number of regions on clustering performance, we implement four StNP systems, which have 4, 8, 16 and 20 regions, respectively. Table 2 lists the comparison results of different number of regions on ten data sets. It is observed from Table 2 that the presented clustering algorithm achieves best clustering performance (i.e., the lowest mean value and the smallest standard deviation) in the case of 20 regions.

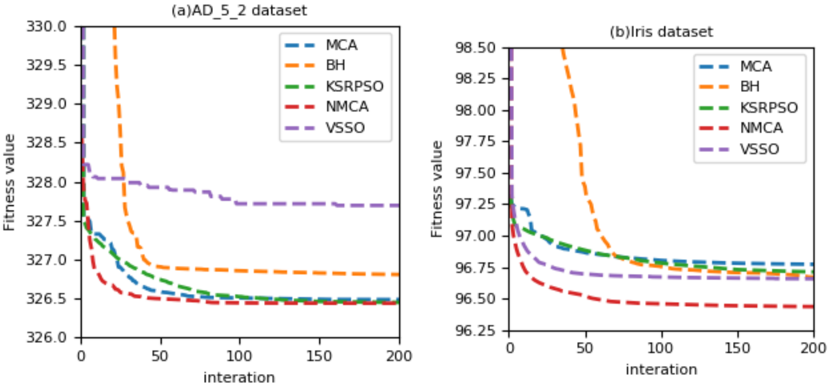


FIGURE 5. Average convergence curves of five clustering algorithms on AD_5_2 and Iris data sets, respectively.

TABLE 3. The comparison results of different algorithms on ten data sets.

Data set	NMCA	MCA	VSSO	BH	KSRPSO	k-means
AD_5_2	**326.44** (**±0.0008**)	**326.44** (±0.0105)	327.67 (±0.2744)	326.99 (±0.0729)	326.46 (±0.0296)	332.47 (±3.1286)
Data_9_2	**590.61** (**±0.00001**)	591.06 (±0.0280)	591.23 (±0.0706)	591.23 (±0.056)	590.71 (±0.1569)	623.57 (±3.1326)
Square_4	**2379.74** (**±0.0005**)	**2379.74** (±0.0189)	2381.65 (±0.0194)	**2379.74** (±0.006)	2379.82 (±0.0061)	2386.00 (±4.5217)
Sym_3_22	**1247.69** (**±3.217e-06**)	1247.72 (±0.0105)	1247.74 (±0.0807)	1247.74 (±0.0557)	1247.76 (±0.1277)	1255.45 (±3.8725)
Iris	**96.39** (±0.0107)	96.75 (±0.0428)	96.66 (**±0.0**)	96.66 (±0.0017)	96.72 (±0.0661)	104.11 (±12.4563)
Wine	**16292.25** (**±0.0031**)	**16292.25** (±0.1529)	16292.76 (±0.82)	16294.32 (±1.6513)	16294.10 (±0.5218)	16312.43 (±9.4269)
Livedisorder	**9851.72** (**±0.0029**)	9851.73 (±0.0347)	9866.64 (±1.9326)	10107.28 (±44.1096)	9910.24 (±23.4412)	9868.32 (±7.9274)
LungCancer	**124.65** (±0.0106)	124.69 (**±0.0011**)	125.60 (±0.4619)	125.12 (±0.3826)	124.91 (±0.2501)	139.40 (±7.3136)
Newthyroid	**1867.15** (**±0.7425**)	1869.29 (±0.9215)	1946.53 (±11.3005)	1946.11 (±9.8419)	1899.72 (±7.7688)	1886.25 (±16.2189)
BreastCancer	2964.41 (±0.1376)	2970.24 (±1.1225)	**2964.39** (**±0.0**)	**2964.39** (±0.00921)	2966.82 (±0.1676)	2981.79 (±10.43651)

4) EXPERIMENTAL RESULTS AND ANALYSIS

After analyzing effects of different parameters on experimental results, we chose an optimal group of priori parameters to realize an StNP system, where it had 20 cells ($m = 20$), each cell had 50 variables ($|V_i| = n = 50, 1 \leq i \leq 20$) and maximum number of iterations was 100. Based on the StNP system, experiments of the presented clustering algorithm on ten data sets have been accomplished and compared with other five clustering algorithms. The comparison results provide mean value and standard deviation of each algorithm in term of J_m.

Figure 4(b) gives the clustering results of the proposed clustering algorithm on four 2-dimensional artificial datasets, while Figure 4(a) shows the corresponding original datasets. The experimental results indicate that the clustering results are more consistent with human vision cognition, and they are also better to meet the clustering criteria: data in the same cluster are closer, while data in different clusters are farther away.

Figure 5 shows average convergence curves of five clustering algorithms for 30 execution on AD_5_2 and Iris data sets, respectively, including NMCA, MCA, VSSO, BH and KSRPSO algorithms. It can be observed from Figure 5 that compared with other four algorithms, NMCA has faster convergence speed on the two data sets. Moreover, NMCA achieves the lowest J_m value (i.e., the best clustering performance) for the two data sets in all the five algorithms. At the same time, we can see that NMCA exceeds MCA (an existing clustering method based on P systems) in terms of both convergence speed and convergence performance. The comparison results indicate that the developed stochastic production function-repartition protocol can enhance the capability of searching for the optimal cluster centers.

To further evaluate the clustering performance of the proposed clustering algorithm, it was compared with five state-of -the-art clustering algorithms based on optimization techniques for ten datasets. Table 3 gives the comparison results of six methods on ten datasets which are mean and standard deviation of 30 executions for each algorithm. For mean value, NMCA achieves the lowest value on each of nine data sets except for BreastCancer. For BreastCancer data set, VSSO and BH achieve the lowest value, 2964.39, but, the value of NMCA is very close to it, i.e., 2964.41. The comparison results demonstrate that the proposed clustering algorithm can achieve the best clustering performance on the ten data sets.

It is found from Table 3 that NMCA achieves smallest standard deviation on each of six datasets, including AD_5_2, Data_9_2, Square_4, Wine, Livedisorder and Newthyroid. We can also observe that: (i) NMCA has second small standard deviation for Sym_3_22; (ii) NMCA has third small standard deviation for Iris; (iii) NMCA has second small standard deviation for LungCancer; (iv) NMCA has second small standard deviation for BreastCancer. The comparison results demonstrate that the proposed clustering algorithm has higher robustness.

V. CONCLUSION AND FUTURE WORK

This paper introduced a novel stochastic production function-repartition protocol and presented an extension of numerical P systems, called stochastic numerical P (StNP) systems. Compared with usual production function-repartition protocol, there are three different characteristics in stochastic production function-repartition protocol:

1. some stochastic/randomized factors are contained in production function and/or repartition protocol;
2. stochastic production function can use the variables in the region where it resides and/or the variables in the following upper (parent) and lower (children) regions, while stochastic repartition protocol can only use local variables in the region where it resides.
3. in stochastic production function-repartition protocol, two data updating ways can be used flexibly.

Therefore, StNP systems possess stochastic and dynamic characteristics.

Data clustering problems were used to evaluate the ability of StNP systems. Based on StNP systems, a novel partition clustering algorithm was developed to deal with the data clustering problems, where a specific stochastic production function-repartition protocol was designed. The clustering algorithm has been evaluated on ten benchmark data sets. The comparison results to other clustering algorithms demonstrate the availability and effectiveness of StNP systems in solving data clustering problems.

StNP systems are a kind of distributed parallel computing models with stochastic and dynamic characteristics. Can StNP systems with stochastic and dynamic characteristics be used for other real applications? Our future work will attempt to use StNP systems in other applications, for example, home energy management problems and control problems in robots.

REFERENCES

[1] G. Paŭn, "Computing with membranes," *J. Comput. Syst. Sci.*, vol. 61, no. 1, pp. 108–143, Aug. 2000.

[2] G. Păun, G. Rozenberg, and A. Salomaa, *The Oxford Handbook of Membrane Computing*. New York, NY, USA: Oxford Univ. Press, 2010.

[3] F. Bernardini and M. Gheorghe, "Population P systems," *J. Univ. Comput. Sci.*, vol. 10, no. 5, pp. 509–539, 2004.

[4] L. Ciencialová, E. Csuhaj-Varjú, A. Kelemenová, and G. Vaszil, "Variants of P colonies with very simple cell structure," *Int. J. Comput., Commun. Control*, vol. 4, no. 3, pp. 224–233, 2009.

[5] R. Freund, G. Păun, and M. J. Pérez-Jiménez, "Tissue-like P systems with channel-states," *Theor. Comput. Sci.*, vol. 330, no. 1, pp. 101–116, 2005.

[6] M. Gheorghe, F. Ipate, R. Lefticaru, M. J. Pérez-Jiménez, A. Ţurcanu, L. V. Cabrera, M. García-Quismondo, and L. Mierlă, "3-Col problem modelling using simple kernel P systems," *Int. J. Comput. Math.*, vol. 90, no. 4, pp. 816–830, Apr. 2013.

[7] M. Ionescu, G. Păun, and T. Yokomori, "Spiking neural P systems," *Fundam. Inf.*, vol. 71, no. 2, pp. 279–308, 2006.

[8] G. Păun and M. J. Pérez-Jiménez, "Solving problems in a distributed way in membrane computing: dP systems," *Int. J. Comput., Commun. Control*, vol. 5, no. 2, pp. 238–250, 2010.

[9] H. Peng, R. Chen, J. Wang, X. Song, T. Wang, F. Yang, and Z. Sun, "Competitive spiking neural P systems with rules on synapses," *IEEE Trans. Nanobiosci.*, vol. 16, no. 8, pp. 888–895, Dec. 2017.

[10] H. Peng, J. Yang, J. Wang, T. Wang, Z. Sun, X. Song, X. Luo, and X. Huang, "Spiking neural P systems with multiple channels," *Neural Netw.*, vol. 95, pp. 66–71, Nov. 2017.

[11] X. Song, J. Wang, H. Peng, G. Ning, Z. Sun, T. Wang, and F. Yang, "Spiking neural P systems with multiple channels and anti-spikes," *Biosystems*, vols. 169–170, pp. 13–19, Jul. 2018.

[12] B. Song, C. Zhang, and L. Pan, "Tissue-like P systems with evolutional symport/antiport rules," *Inf. Sci.*, vol. 378, pp. 177–193, Feb. 2017.

[13] H. Peng, J. Wang, M. J. Pérenz-Jiménez, and A. Riscos-Núñez, "Dynamic threshold neural P systems," *Knowl.-Based Syst.*, vol. 163, pp. 875–884, Jan. 2019.

[14] H. Peng and J. Wang, "Coupled neural P systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1672–1682, Jun. 2019.

[15] H. Peng, B. Li, J. Wang, X. Song, T. Wang, L. Valencia-Cabrera, I. Pérez-Hurtado, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "Spiking neural P systems with inhibitory rules," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105064.

[16] H. Peng, Z. Lv, B. Li, X. Luo, J. Wang, X. Song, T. Wang, M. J. Pérenz-Jiménez, and A. Riscos-Núñez, "Nonlinear spiking neural P systems," *Int. J. Neural Syst.*, 2020, doi: 10.1142/S0129065720500082.

[17] G. Păun and M. J. Pérez-Jiménez, "Membrane computing: Brief introduction, recent results and applications," *Biosystems*, vol. 85, no. 1, pp. 11–22, Jul. 2006.

[18] H. Peng, J. Wang, M. J. Pérez-Jiménez, H. Wang, J. Shao, and T. Wang, "Fuzzy reasoning spiking neural P system for fault diagnosis," *Inf. Sci.*, vol. 235, pp. 106–116, Jun. 2013.

[19] J. Wang, P. Shi, H. Peng, M. J. Perez-Jimenez, and T. Wang, "Weighted fuzzy spiking neural P systems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 2, pp. 209–220, Apr. 2013.

[20] L. Huang, I. H. Suh, and A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Inf. Sci.*, vol. 181, no. 11, pp. 2370–2391, Jun. 2011.

[21] G. Zhang, H. Rong, F. Neri, and M. J. Pérez-Jiménez, "An optimization spiking neural P system for approximately solving combinatorial optimization problems," *Int. J. Neural Syst.*, vol. 24, no. 5, pp. 1–16, 2014.

[22] D. Díaz-Pernil, A. Berciano, F. Peña-Cantillana, and M. A. Gutiérrez-Naranjo, "Segmenting images with gradient-based edge detection using membrane computing," *Pattern Recognit. Lett.*, vol. 34, no. 8, pp. 846–855, Jun. 2013.

[23] H. Peng, J. Wang, and M. J. Pérez-Jiménez, "Optimal multi-level thresholding with membrane computing," *Digit. Signal Process.*, vol. 37, pp. 53–64, Feb. 2015.

[24] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "The framework of P systems applied to solve optimal watermarking problem," *Signal Process.*, vol. 101, pp. 256–265, 2014.

[25] J. Wang, P. Shi, and H. Peng, "Membrane computing model for IIR filter design," *Inf. Sci.*, vol. 329, pp. 164–176, Feb. 2016.

[26] B. Alsalibi, I. Venkat, and M. A. Al-Betar, "A membrane-inspired bat algorithm to recognize faces in unconstrained scenarios," *Eng. Appl. Artif. Intell.*, vol. 64, pp. 242–260, Sep. 2017.

[27] H. Peng, J. Wang, J. Ming, P. Shi, M. J. Perez-Jimenez, W. Yu, and C. Tao, "Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4777–4784, Sep. 2018.

[28] H. Peng, J. Wang, P. Shi, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Fault diagnosis of power systems using fuzzy tissue-like P systems," *ICA*, vol. 24, no. 4, pp. 401–411, Sep. 2017.

[29] T. Wang, G. Zhang, J. Zhao, Z. He, J. Wang, and M. J. Perez-Jimenez, "Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1182–1194, May 2015.

[30] M. García-Quismondo, M. Levin, and D. Lobo, "Modeling regenerative processes with membrane computing," *Inf. Sci.*, vol. 381, pp. 229–249, Mar. 2017.

[31] M. García-Quismondo, I. C. T. Nisbet, C. Mostello, and J. M. Reed, "Modeling population dynamics of roseate terns (Sterna dougallii) in the Northwest Atlantic Ocean," *Ecol. Model.*, vol. 368, pp. 298–311, Jan. 2018.

[32] M. Gheorghe, V. Manca, and F. J. Romero-Campero, "Deterministic and stochastic P systems for modelling cellular processes," *Natural Comput.*, vol. 9, no. 2, pp. 457–473, Jun. 2010.

[33] G. Păun and R. Păun, "Membrance computing and economics: Numberic P systems," *Fundam. Inf.*, vol. 73, nos. 1–2, pp. 213–227, 2006.

[34] A. Pavel, O. Arsene, and C. Buiu, "Enzymatic numerical P systems—A new class of membrane computing systems," in *Proc. IEEE 5th Int. Conf. Bio-Inspired Comput., Theories Appl. (BIC-TA)*, Sep. 2010, pp. 1331–1336.

[35] Z. Zhang and L. Pan, "Numerical P systems with thresholds," *Int. J. Comput. Commun. Control*, vol. 11, no. 2, p. 292, Jan. 2016.

[36] L. Pan, Z. Zhang, T. Wu, and J. Xu, "Numerical P systems with production thresholds," *Theor. Comput. Sci.*, vol. 673, pp. 30–41, Apr. 2017.

[37] L. Liu, W. Yi, Q. Yang, H. Peng, and J. Wang, "Numerical P systems with Boolean condition," *Theor. Comput. Sci.*, vol. 785, pp. 140–149, Sep. 2019.

[38] Z. Zhang, T. Wu, A. Păun, and L. Pan, "Numerical P systems with migrating variables," *Theor. Comput. Sci.*, vol. 641, pp. 85–108, Aug. 2016.

[39] C. Buiu, O. Arsene, C. Cipu, and M. Patrascu, "A software tool for modeling and simulation of numerical P systems," *Biosystems*, vol. 103, no. 3, pp. 442–447, Mar. 2011.

[40] C. Buiu, C. Vasile, and O. Arsene, "Development of membrane controllers for mobile robots," *Inf. Sci.*, vol. 187, pp. 33–51, Mar. 2012.

[41] A. B. Pavel and C. Buiu, "Using enzymatic numerical P systems for modeling mobile robot controllers," *Natural Comput.*, vol. 11, no. 3, pp. 387–393, Sep. 2012.

[42] X. Wang, G. Zhang, F. Neri, T. Jiang, J. Zhao, M. Gheorghe, F. Ipate, and R. Lefticaru, "Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots," *Integr. Comput.-Aided Eng.*, vol. 23, no. 1, pp. 15–30, Dec. 2015.

[43] C. Buiu and A. G. Florea, "Membrane computing models and robot controller design, current results and challenges," *J. Membrane Comput.*, vol. 1, no. 4, pp. 262–269, Dec. 2019.

[44] A. G. Florea and C. Buiu, "Sensor fusion for autonomous drone waypoint navigation using ROS and numerical P systems: A critical analysis of its advantages and limitations," in *Proc. 22nd Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2019, pp. 112–117.

[45] A. Leporati, A. E. Porreca, C. Zandron, and G. Mauri, "Improving universality results on parallel enzymatic numerical P systems," *Int. J. Unconventional Comput.*, vol. 9, no. 5, pp. 385–404, 2013.

[46] C. I. Vasile, A. B. Pavel, and I. Dumitrache, "Universality of enzymatic numberical P systems," *Int. J. Comput. Math.*, vol. 90, no. 4, pp. 869–879, 2013.

[47] C. I. Vasile, A. B. Pavel, I. Dumitrache, and G. Păun, "On the power of enzymatic numerical P systems," *Acta Inf.*, vol. 49, no. 6, pp. 395–412, 2012.

[48] Z. Zhang, T. Wu, and L. Pan, "On string languages generated by sequential numerical P systems," *Fundam. Inf.*, vol. 145, no. 4, pp. 485–509, Aug. 2016.

[49] Z. Zhang, T. Wu, A. Păun, and L. Pan, "Universal enzymatic numerical P systems with small number of enzymatic variables," *Sci. China Inf. Sci.*, vol. 61, no. 9, Sep. 2018.

[50] M. C. Naldi, R. J. G. B. Campello, E. R. Hruschka, and A. C. P. L. F. Carvalho, "Efficiency issues of evolutionary k-means," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1938–1952, 2011.

[51] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 183–197, Jan. 2010.

[52] D. Chang, X. Zhang, and C. Zheng, "A genetic algorithm with gene rearrangement for k-means clustering," *Pattern Recognit.*, vol. 42, pp. 1210–1222, Jul. 2009.

[53] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, "Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 3196–3206, Jun. 2013.

[54] U. Maulik and I. Saha, "Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery," *Pattern Recognit.*, vol. 42, no. 9, pp. 2135–2149, Sep. 2009.

[55] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 652–657, Jan. 2011.

[56] J. Xue and X. Liu, "Lattice based communication P systems with applications in cluster analysis," *Soft Comput.*, vol. 18, no. 7, pp. 1425–1440, Jul. 2014.

[57] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "An unsupervised learning algorithm for membrane computing," *Inf. Sci.*, vol. 304, pp. 80–91, May 2015.

[58] H. Peng, Y. Jiang, J. Wang, and M. J. Pérez-Jiménez, "Membrane clustering algorithm with hybrid evolutionary mechanisms," *J. Softw.*, vol. 26, no. 5, pp. 1001–1012, 2015.

[59] H. Peng, J. Wang, P. Shi, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "An extended membrane system with active membrane to solve automatic fuzzy clustering problems," *Int. J. Neural Syst.*, vol. 26, pp. 1–17, May 2016.

[60] H. Peng, J. Wang, P. Shi, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "An automatic clustering algorithm inspired by membrane computing," *Pattern Recognit. Lett.*, vol. 68, pp. 34–40, 2015.

[61] H. Peng, P. Shi, J. Wang, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "Multiobjective fuzzy clustering approach based on tissue-like membrane systems," *Knowl.-Based Syst.*, vol. 125, pp. 74–82, Jun. 2017.

[62] X. Liu and J. Xue, "A cluster splitting technique by hopfield networks and P systems on simplices," *Neural Process. Lett.*, vol. 46, no. 1, pp. 171–194, Aug. 2017.

[63] S. Yan, Y. Wang, D. Kong, J. Hu, J. Qu, X. Liu, and J. Xue, "Hybrid chain-hypergraph P systems for multiobjective ensemble clustering," *IEEE Access*, vol. 7, pp. 143511–143523, 2019, doi: 10.1109/access.2019.2944675.

[64] *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

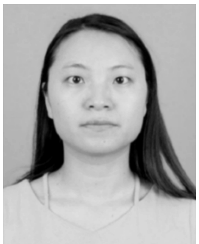
[65] H. Peng, X. Luo, Z. Gao, J. Wang, and Z. Pei, "A novel clustering algorithm inspired by membrane computing," *Sci. World J.*, vol. 2015, pp. 1–8, Mar. 2015.

[66] W.-C. Yeh and C.-M. Lai, "Accelerated simplified swarm optimization with exploitation search scheme for data clustering," *PLoS ONE*, vol. 10, no. 9, Sep. 2015, Art. no. e0137246.

[67] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013.

[68] C.-Y. Tsai and I.-W. Kao, "Particle swarm optimization with selective particle regeneration for data clustering," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6565–6576, Jun. 2011.

[69] E. Forgy, "Cluster analysis of multivariate data: Efficicncy versus interpretability of classification," *Biometrics*, vol. 2, pp. 768–769, 1965.



JINYU YANG received the B.Sc. degree in software engineering and the M.Sc. degree in computer science from Xihua University, Chengdu, China, in 2016 and 2019, respectively. She is currently pursuing the Ph.D. degree with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Her current research interests include image and signal processing, membrane computing, and machine learning.

HONG PENG (Member, IEEE) received the B.Sc. degree and the M.E. degree in mathematics from Sichuan Normal University, Chengdu, China, in 1987 and 1990, respectively, and the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2010. He was a Lecturer with the Sichuan College of Science and Technology, China, from 1990 to 1999, and an Associate Professor with Xihua University, China, from 2000 to 2004, where he has been a Professor with the School of Computer and Software Engineering, since 2005. He was a Visiting Scholar with the Research Group of Natural Computing, University of Seville, Spain, from 2011 to 2012. He has published over 100 scientific articles in international journals. His research interests include membrane computing, machine learning, pattern recognition, image processing, and computer vision.

XIAOHUI LUO received the Ph.D. degree in instrument science and technology from Chongqing University, Chengdu, China, in 2002. He is currently an Associate Professor with the School of Computer and Software Engineering, Xihua University, China. His research interests include pattern recognition, image processing, computer vision, and machine learning.

JUN WANG received the Ph.D. degree in electrical engineering from Southwest Jiaotong University, China, in 2006. She was a Lecturer with the Sichuan College of Science and Technology, China, from 1991 to 2003. She was an Associate Professor with Xihua University, China, from 1998 to 2003. She has been a Professor with the School of Electrical Engineering and Electronic Information, since 2004. Her research interests include electrical automation, intelligent control, and membrane computing.