

Article

A Grid-Density Based Algorithm by Weighted Spiking Neural P Systems with Anti-Spikes and Astrocytes in Spatial Cluster Analysis

Deting Kong, Yuan Wang, Xinyan Wu, Xiyu Liu, Jianhua Qu * and Jie Xue *

Business School, Shandong Normal University, East Road of Wenhua, No.88, Jinan 250014, China; detkong@163.com (D.K.); wantedumi@163.com (Y.W.); wxy1325827437@163.com (X.W.); sdxyliu@163.com (X.L.)

* Correspondence: qjh@sdnu.edu.cn (J.Q.); Jiexue@sdnu.edu.cn (J.X.)

Received: 18 July 2020; Accepted: 4 September 2020; Published: 11 September 2020



Abstract: In this paper, we propose a novel clustering approach based on P systems and grid-density strategy. We present grid-density based approach for clustering high dimensional data, which first projects the data patterns on a two-dimensional space to overcome the curse of dimensionality problem. Then, through meshing the plane with grid lines and deleting sparse grids, clusters are found out. In particular, we present weighted spiking neural P systems with anti-spikes and astrocyte (WSNPA2 in short) to implement grid-density based approach in parallel. Each neuron in weighted SN P system contains a spike, which can be expressed by a computable real number. Spikes and anti-spikes are inspired by neurons communicating through excitatory and inhibitory impulses. Astrocytes have excitatory and inhibitory influence on synapses. Experimental results on multiple real-world datasets demonstrate the effectiveness and efficiency of our approach.

Keywords: spiking neural p systems; grid-density based clustering approach; multidimensional datasets

1. Introduction

Spiking neural P systems (SN P in short) are a kind of parallel and distributed neural-like computation model in the field of membrane computing [1,2]. SN P systems, which are inspired by neural cells [3], have a series of spikes and information processing rules, called firing and forgetting rules [4]. Inspired by different biological phenomena and mathematical motivations, several families of SN P systems have been constructed, such as SN P systems with anti-spikes [5], SN P systems with weight [6], SN P systems with astrocyte [7], stochastic numerical P systems [8], SN P systems with threshold [9], numerical spiking neural P systems [10], double layers self-organized spiking neural P systems [11], SN P systems with rules on synapses [12], SN P systems with structural plasticity [13]. For applications, SN P systems are used to design logic gates, logic circuits [7] and operating systems [14], perform basic arithmetic operations [15], solve combinatorial optimization problems [16], and realize fingerprint recognition [11]. Păun who initiated the P systems pointed out that solving real problems by membrane computing needs to be addressed [17]. The comparative analysis of dynamic behaviors of a hybrid algorithm indicates that the combination of evolutionary computation with P systems can produce a better algorithm for balancing exploration and exploitation [18–20]. However, the hybrid algorithm does not use objects and rules defined by P systems. On account of the P system is still in the phase of solving addition, subtraction, multiplication, and division [13]. How does the P system realize more complex and universally applicable functions? Clustering algorithm has universal applicability, and its inherent characteristics make it especially suitable for parallel operation through P system to realize the possibility of reducing time complexity. The whole process of clustering algorithm proposed in this paper is implemented through changes of

objects by rules in membranes. In which, objects encode data. Membrane rules working on objects achieve the clustering goal. Real-world datasets always have multiple attributes, so these datasets often include high-dimensional data with features of multiple dimensions. Grid-based clustering is usually used for the more complex and high-dimension data. Data space is partitioned into certain number of cells. Cells are basic units for clustering operations [21]. OPTIGRID [22] is designed to obtain an optimal grid partitioning. CLIQUE is probably the most intuitive and comprehensive clustering technique [23]. The shifting grid approach (SHIFT) has been reported to be somehow similar to the sliding window technique. However, grid-based clustering methods face the curse of dimensionality. In other words, as the dimensionality increases, the number of grids increases exponentially. In order to solve this problem, some methods proposed to select two features to form a plane before meshing. In [24], random projection is used to reduce the dimensionality of the data. The dynamic feature mask is proposed to deal with the feature selection problem [25]. However, the features selected through these methods are not always the most distinguishable. In order to further improve the clustering effect, we propose to select features based on the data distribution histogram of each dimension. Inspired by AGRID [26], we combine the grid-based clustering method with the density-based clustering method. Based on the above considerations, this paper develops a hybrid optimization method, grid-density based algorithm by weighted SN P systems with anti-spike and astrocyte. Characteristic of each dimension is calculated and compared by rules independently in different membranes synchronously. Communications among membranes is utilized to explore clusters. Experimental results on multiple real-world datasets demonstrate the effectiveness and efficiency of our approach.

2. Methods

2.1. Weighted Spiking Neural P Systems with Anti-Spikes and Astrocytes

Weighted spiking neural P systems with anti-spikes and astrocytes (called WSNPA2) of degree $m \geq 1$ is a construct of the form

$$\Pi = (O, \sigma_1, \dots, \sigma_m, syn, ast_1, \dots, ast_k, In, Out)$$

where, O is the set of spikes, $O = \{a, \bar{a}\}$, a is spike, \bar{a} is anti-spike. The empty string is denoted by λ ; $\sigma_1, \sigma_2, \dots, \sigma_m$ are neurons, m is the degree of neurons, of the form $\sigma_i = (n_i, R_i)$, $1 \leq i \leq m$, Where, n_i is the initial number of spikes contained in σ_i , R_i is a finite set of rules with: (1) $E/s^c \rightarrow s$, s is spikes or anti-spikes, c is the number of spikes in the rule, $c \geq 1$, E is a regular expression over a or \bar{a} ; (2) $s^e \rightarrow \lambda$, e is the number of spikes, $e \geq 1$. $syn \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\} \times \omega$ are synapses between neurons, ω is the weight on synapse (i, j) , $\omega \in \mathbb{Z}$. For each (i, j) , there is at most one synapse (i, j, ω) . A rule $E/s^c \rightarrow s$ is applied as follows. If neuron σ_i contains r spikes/anti-spikes, $r \geq c$, then the rule can fire, c numbers of spikes/anti-spikes are consumed, $r - c$ numbers of spikes/anti-spikes remain in σ_i and one spikes/anti-spikes is released. The number of spikes/anti-spikes is multiplied by ω and pass immediately to all neurons with $(i, j, \omega) \in syn$. $s^e \rightarrow \lambda$ is forgetting rules. e numbers of spikes/anti-spikes are omitted from the neuron immediately.

For spikes a^q and anti-spikes a^p ($p, q \in \mathbb{Z}$ are numbers of spikes and anti-spikes), an annihilation rule $a\bar{a} \rightarrow \lambda$ is applied in a maximal manner. a^{q-p} or $a(a)^{p-q}$ remain for the next step, provided that $q \geq p$ or $p \geq q$, respectively. ast_1, \dots, ast_k are astrocytes, of the form $ast_i = (syn_{ast_i}, t_i)$, where $syn_{ast_i} \subseteq syn$ is the subset of synapses controlled by the astrocyte, t_i is the threshold of the astrocyte. Suppose that there are k spikes passing along the neighboring synapses syn_{ast_i} . If $k \geq t_i$, then ast_i has an inhibitory influence on syn_{ast_i} , and the k spikes are transformed into one spike by $a^k \rightarrow a$. a will be sent into the neuron connected to ast_i . Otherwise, $k < t_i$, then ast_i has an excitatory influence on syn_{ast_i} , all spikes survive and reach their destination neurons.

$In, Out \in \{1, 2, \dots, m\}$ indicate the input and output neurons, respectively.

2.2. Grid-Density Based Clustering Algorithm for Multidimensional Dataset

2.2.1. Identify the Two Well-Informed Features

Generally, in grid-based methods, the computations will grow exponentially with high dimensions, because of the evaluations should be done over all grid points. For example, a cluster analysis with N dimensions and L grid partitions in each dimension, would result in L^N grids. To avoid this curse of dimensionality problem, we try to project data in actual feature space into a 2D space, aim to discover the initial locations of spike clusters in a plane. The plane comprised by the two well-informed features $n_i, n_j \in N$ will be covered by a $L \times L$ lattice of grids with M data objects $X_p (p = 1, \dots, M)$.

At first, each dimension of objects is partitioned into $K = \lceil \sqrt{M} \rceil$ bins, $B = \{b_1, b_2, \dots, b_k\}$ is as

$$c_i(b_k) = |\{X_p, p = 1, \dots, M, | X_{pi} - b_k| < | X_{pi} - b_{k'}|; b_k, b_{k'} \in B, b_{k'} \neq b_k\}| \quad (1)$$

x_{pi} is the value of feature n_i in data pattern X_p and $|\cdot|$ is the cardinality operator representing the number of elements in a set. For each attribute of the Wine data set, we draw a histogram according to the above rules. Since the data set has 13 attributes, we get 13 feature maps. Figure 1 depicts this histogram for the 13 features of the known Wine data set. Because the number of peaks means the ability to divide the data in this data set by this feature, we take ε , which represents the number of peaks, as the measurement standard. As is shown in Figure 1, the ε in these histograms are 3, 2, 1, 2, 1, 2, 2, 4, 3, 3, 3, 4, 5, respectively. According to these values of ε , the features n_8, n_{12} and n_{13} are selected. If there are same maximum values of ε , we divide each dimension of the data object into $K/2$ bins to recalculate the number of peaks until two well-informed features are selected. These features will then be used to do the cluster analysis.

$$\varepsilon(n_i) = |\{c_i(b_k) : c_i(b_k) > c_i(b_{k-1}) \text{ and } c_i(b_k) > c_i(b_{k+1})\}| \quad (2)$$

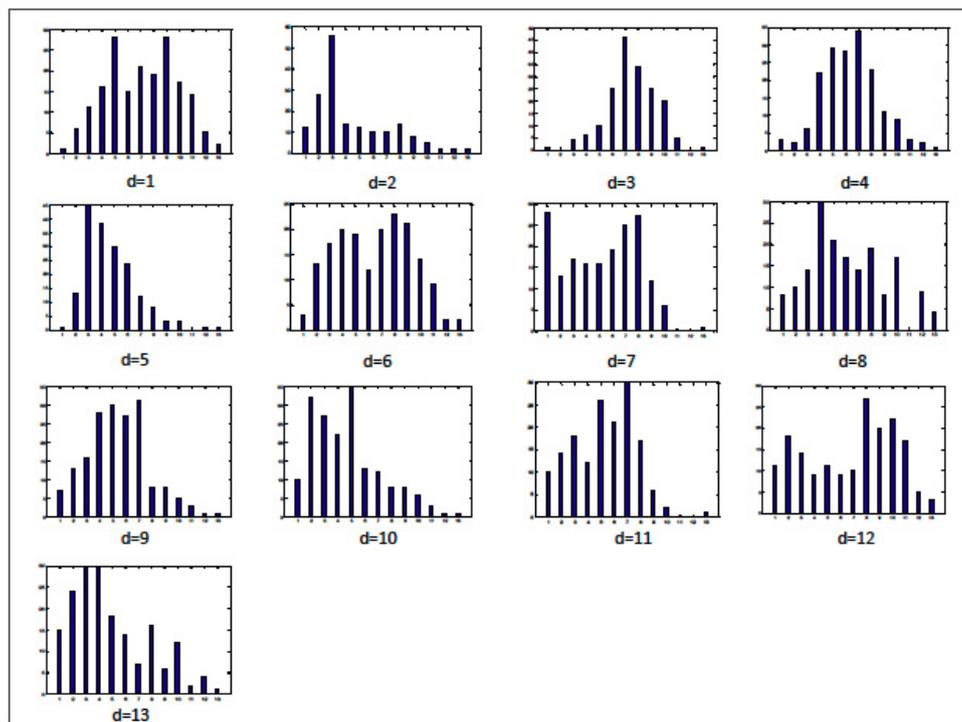


Figure 1. histogram for the 13 features of Wine data set.

2.2.2. Clustering by Grid-Density Based Algorithm

The plane comprised by the two well-informed features will be covered by a $H = L \times L$ lattice of grids. Grids are denoted by $G = \{g_1, g_2, \dots, g_H\}$. $C(g_h), h \in \{1, 2, \dots, H\}$ is the number of data X_p partitioned in grid g_h according to (3).

$$C(g_h) = |\{X_p : p = 1, \dots, M, X_p \in g_h, g_h \in G\}| \quad (3)$$

Next, non-dense grids are deleted. A grid is dense if $C(g_h) > \theta, \theta \in N^+$ is a threshold defined before computation. The threshold is initialized to 2% of the number of data, and on this basis, it floats upwards by 10% and downwards by 1%. Several experiments are performed to select the threshold that makes the clustering effect the best. After getting the initial members of grid graph G , G is refined by finding out dense grid. Those sparse grids are discarded. The refined grid graph is defined as:

$$G_r = \{g_h \mid C(g_h) > \theta\} \subseteq G \quad (4)$$

Each grid $g_h \in G_r$ has 4 neighbors connected with it as shown in Figure 2. When there is no dense grid in the cluster that can be connected to the grid, a cluster is formed. A cluster is a set of neighbors of dense grids. The process of clustering algorithm is shown in Algorithm 1 below.

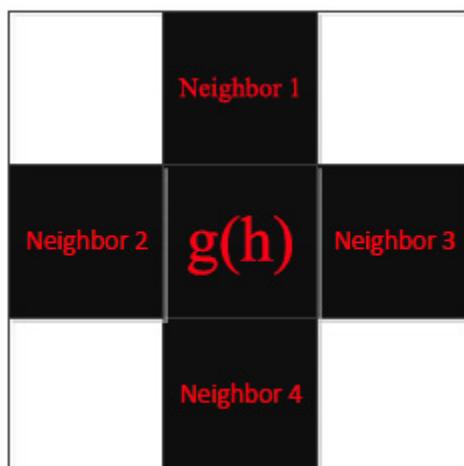


Figure 2. Neighbors of grid $g_h \in G_r$.

Algorithm 1: Algorithm: Grid-Density based clustering algorithm.

Inputs: $\Omega = \{X_{pi}, 1 \leq p \leq M, 1 \leq i \leq N\}, H = L \times L, \theta : \text{densitythreshold}$

Outputs: $CS = \{CS_1, CS_2, \dots, CS_t\}$

Begin

for all features $n_i, i = 1, 2, \dots, N$

use $K = \lceil \sqrt{M} \rceil$ bins to partition the feature n_i

obtain the number of data in each bin $B = \{b_1, \dots, b_k\}$ by (1)

compute the effectiveness measure $\varepsilon(n_i)$ for n_i by (2)

rank $\varepsilon(n_i)$

get the two top-ranked feature s

project data patterns into $H = L \times L$ grid s

obtain the capability $C(g_H)$ of each grid by (3)

select dense grid by (4)

form cluster set by combing neighbor dense grids

return the t clusters, $CS = \{CS_1, CS_2, \dots, CS_t\}$

End

2.3. Multi-WSNPA2 Design for Grid-Density Based Clustering

2.3.1. Grid-Density Based Clustering by Multi-WSNPA2

In this section, the weighted spiking neural P system with anti-spikes and astrocytes is designed for grid-density based clustering. Objects in each neuron are organized as spikes and anti-spikes with real-valued numbers corresponding to $\Omega = \{X_{pi}, 1 \leq p \leq M, 1 \leq i \leq N\}$. Feature selection and cluster analysis are implemented by rules of WSNPA2. WSNPA2 is divided into three subsystems: feature selection, effectiveness comparison and clustering. The structure of WSNPA2 is shown in Figure 3, where ovals represent neurons, rhombic stand for astrocytes and arrows indicate channels. WSNPA2 for grid-density based clustering algorithm is described as the following construct

$$\Pi = (O, \sigma_{S1}, \sigma_{S2}, \sigma_{S3}, syn, R, ast_{S1}, ast_{S3}, \sigma_{in1}, \dots, \sigma_{inN}, \sigma_{out1}, \dots, \sigma_{out_t})$$

where, $O = \{a, \bar{a}\}$. At beginning, the input neuron contains x_{pi} numbers of spike a ; σ_{S1} stands for neurons in feature selection subsystems, $\sigma_{S1} = \{DM_{iz}, Fi_{z'}, FS_{iz}''\}, 1 \leq i \leq N, 1 \leq z = z' \leq \lfloor \sqrt{M} \rfloor, 1 \leq z'' \leq 2\lfloor \sqrt{M} \rfloor / 3$; σ_{S2} represents neurons in effectiveness comparison subsystems, $\sigma_{S2} = \bigcup_{1 \leq i \leq N} (E_i \cup EC_i) \cup \{ECS\}$; σ_{S3} describes neurons in clustering subsystems, $\sigma_{S3} = \{\{C_{i'}\}, i' \in \{1, 2\}, 1 \leq j \leq N\}, \{G_{gg'}\}, 1 \leq g, g' \leq L\}, CS\}$; The number of astrocytes ast_{S1} in the feature selection system is $N * N$ between each two DM_{iz} . The number of astrocytes ast_{S3} in the clustering system is $L \times L \times 2 + 1$; Input neurons $\sigma_{in1}, \dots, \sigma_{inN}$ are in the feature selection system. Output neurons $\sigma_{out1}, \dots, \sigma_{out_t}$ are in the clustering system.

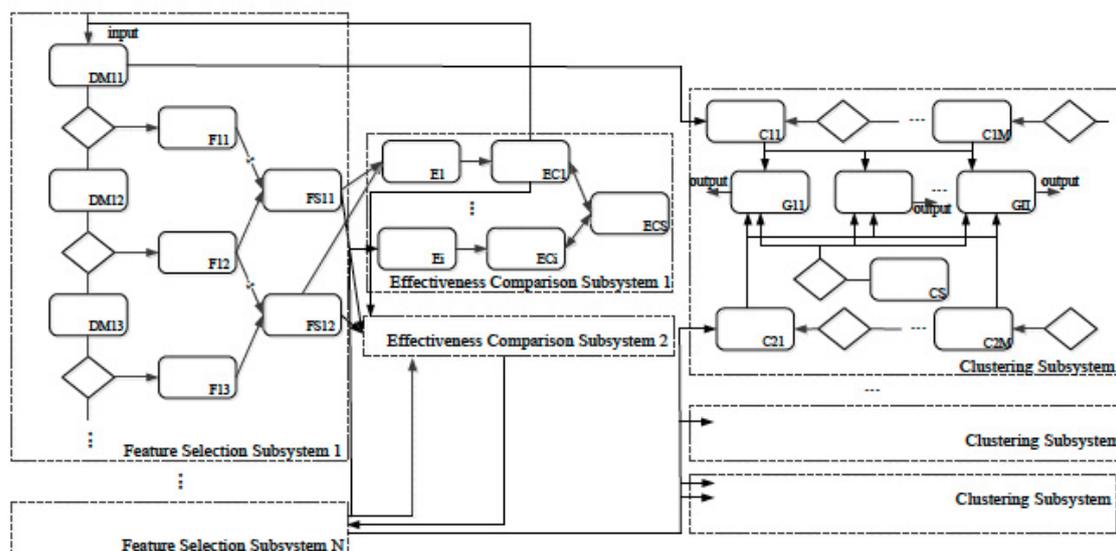


Figure 3. Structure of WSNPA2 for grid-density based clustering algorithm.

There are several different clustering subsystems working in parallel for different grid number $H = L * L$, which means the whole system can output variant clustering results simultaneously. Then, the clustering results obtained by different clustering subsystems are connected according to the neighboring relationship of grid positions, so as to obtain the final clustering result. Multi-WSNPA2 makes the calculation proceed in parallel in feature selection subsystem, effectiveness comparison subsystem and clustering subsystem, respectively. The complexity is reduced from $O(n)$ to $O(kn)$, where k is a constant less than 1. The detail how the complexity of grid-density based algorithm is calculated is as follows:

1. The complexity of traversing N data to form feature histograms is N .
2. The complexity of calculating the amount of data falling into each interval in the histogram is N .

3. The complexity of determining whether the amount of data in each rectangle in the histogram is greater than the left and right sides is K , where K is the number of rectangles.
4. The complexity of finding the two features with the most peaks is A , where A is the number of features in the data set.
5. The complexity of projecting data patterns into $H = L \times L$ grids is N .
6. The complexity of calculating the amount of data in each grid is N .
7. The complexity of selecting dense grid is L^2 .
8. The complexity of combing neighbor dense grids is $L^2 - D$, where D is the number of grids removed.

The complexity of grid-density based algorithm is $O(N + N + K + A + N + N + L^2 + L^2 - D)$, where K , A , L and D are constants. The simplification is $O(n)$. When we use multi-WSNPA2 to calculate the above algorithm, the data traversal in 1, 5, interval traversal in 2 and grid traversal in 6, 7 are parallel operations. So its complexity is $O(1 + \maxdataK + K + K + 1 + \maxdataL + 1 + 1)$, where \maxdataK and \maxdataL are the max value in interval K and max value in grid L , respectively. And the simplification is $O(kn)$, where k is a constant less than 1.

syn represents synapse among neurons:

$$syn(DM_{iz}, ast_{S1}), 1 \leq i \leq N, 1 \leq z \leq [\sqrt{M}]$$

$$syn(F_{iz'}, ast_{S1}), 1 \leq i \leq N, 1 \leq z' \leq [\sqrt{M}]$$

$$syn(F_{iz'}, F S_{iz''}), 1 \leq i \leq D, 1 \leq z' \leq 1 \leq z'' \leq 2[\sqrt{M}]/3$$

$$syn(F S_{iz''}, E_i), 1 \leq i \leq N, 1 \leq z'' \leq 2[\sqrt{M}]/3$$

$$syn(E_i, EC_i), 1 \leq i \leq N$$

$$syn(EC_i, ECS), 1 \leq i \leq N$$

$$syn(DM_{i1}, C_{i1}), 1 \leq i \leq N$$

$$syn(C_{i'j}, ast_{S3}), i' \in \{1, 2\}, 1 \leq j \leq N$$

$$syn(C_{i'j}, G_{gg'}), i' \in \{1, 2\}, 1 \leq j \leq N, 1 \leq g, g' \leq L$$

$$syn(G_{gg'}, ast_{S3}), i' \in \{1, 2\}, 1 \leq g, g' \leq L$$

$$syn(ast_{S3}, CS)$$

R is the following set of firing and forgetting rules: ($[]_x$ means the rule works in neuron x , otherwise, the rule executes through all neurons)

$$[a^{x_{pi}} \rightarrow a^{x_{pi}}, x_{pi} < t_{ih}]_{DM_{iz}}, a^{x_{pi}} \rightarrow a, x_{pi} > t_{ih}$$

$$[a^f \rightarrow a^f]_{F_{iz'}}, [a^{f_2 - f_1} \rightarrow a]_{F S_{iz''}}, f_2 - f_1 > 0$$

$$[a^{2m} \rightarrow a^{2m+2}]_{E_i}, [a^{2m+2}/a^m \rightarrow a^m]_{E_i}, [a^m \rightarrow a^m]_{EC_i}$$

$$[a \rightarrow a]_{E_i}, [a \rightarrow a]_{EC_i}, [a^m \rightarrow \bar{a}^2]_{ECS}, [\bar{a}^2 a^m \rightarrow \lambda]_{E'_i}$$

$$[a^m \rightarrow \bar{a}^2]_{ECS}, [\bar{a}^2 a^m \rightarrow \lambda]_{E'_i}, [\bar{a}^2 a^{x_{ij}} \rightarrow a^{x_{ij}}]_{DM_{iz}}$$

$$[a^{x'_{ij}} \rightarrow a^{x'_{ij}}]_{CS_{i'j}}, a^{x'_{ij}} \rightarrow a, x'_{ij} > \theta_{i'}$$

$$[a \rightarrow a]_{G_{gg'}}, [a^4/a^3 \rightarrow \bar{a}^2]_{G_{gg'}}, [a^n \rightarrow \lambda; n < \theta]_{G_{gg'}}$$

$$[a^n \rightarrow a; n \geq \theta]_{G_{gg'}}, [a \rightarrow a]_{G_{gg'}}$$

2.3.2. Overview of Computations

Data set of M observations are codified by spikes $a^{x_{pi}}, 1 \leq i \leq N, 1 \leq p \leq M$. The computation of the P system is split in three subsystems. When $a^{x_{pi}}$ arrive in neuron DM_{i1} , the computation begins in parallel.

In feature selection subsystem, threshold t_{ih} in each astrocytes ast_{S1ih} is $t_{ih} = [h * (X_{i \max} - X_{i \min}) / \sqrt{M}]$, $1 \leq i \leq N, 1 \leq h \leq \lfloor \sqrt{M} \rfloor$. If $x_{pi} > t_{ih}$, it is said that x_{pi} belongs to the current neuron DM_{iz} . Rule 2 add a spike in DM_{iz} . Otherwise, $a^{x_{pi}}$ pass through DM_{iz} to find its neuron (bin) by rule 1. After all $a^{x_{pi}}$ execute with rule 1 and rule 2, the peak of each dimension is chosen by rule 3 and rule 4.

All peaks of dimension i gain by spike a in neuron E_i . Then, effectiveness comparison subsystem starts. The maximum number of peaks of each dimension is selected by rule 5–9. Rule 5 and 6 copy peaks a^m into a^{2m+2} and sends a^m into neuron EC_i for preparation. Then, different number of a^m is descended one by one by rule 8. Rule 9 helps ECS collect all dimensions without the one with maximum number of peaks. The serial number of the neuron who sends out \bar{a}^2 by rule 10 is chosen as the first dimension for clustering. Other effectiveness comparison subsystem will work in the same way except that the chosen dimension is deleted by rule 11.

Rule 12 activates the input neurons of the two selected features. The clustering subsystem begins. Rule 13–14 put observations into suitable bins in their own dimensions. ($\theta_i = [i^1 X_{i \max} - X_{i \min}) / L]$). Then, rule 15 select the grid who has two spikes. It is chosen as initial grid for cluster. Rule 16 activates the input neurons of the two selected features again. Rule 17–19 finds dense grids. Rule 12–16 will continue to work until there are no spike input. The clustering result is obtained by the serial number of neurons with a output by rule 19.

3. Results and Discussion

The experiments set out to investigate the performance of the proposed approach compared to classical clustering algorithms. We conduct experiments using ten real-world datasets and all datasets are from UCI (<https://archive.ics.uci.edu/ml/datasets.php>). Table 1 summarizes these data sets, ordered in their number of attributes.

The amount of necessary resources to define multi-WSNPA2 of grid-density based clustering for the ten datasets are shown in Table 2.

To compare the algorithm with k-means, AHC (agglomerative hierarchical clustering) and two other new algorithms in more precise notion, their clustering performance in terms of accuracy is depicted in Table 3. This AHC uses the ward linkage²⁷ which is appropriate for Euclidean distance. The accuracy of clusters evaluates the right objects of clusters in each class.

Clearly, the accuracy is comparable to k-means, AHC and two other new algorithms and even better as its averages (in bold-face) show. This means that the clustering effect of our method is better than other algorithms.

Table 1. Ten real-world datasets of UCI.

Data Set	Number of Attributes	Number of Classes	Number of Objects
Haberman	3	2	306
Iris	4	3	150
Thyroid	5	4	215
Ecoli	7	8	336
Diabetes	8	3	768
Breast	9	3	699
Glass	9	6	214
Wine	13	3	178
Vehicle	18	4	846
Ionosphere	34	2	351

Table 2. The amount of necessary resources to define multi-WSNPA2 of the ten datasets.

Data Set	Parallel Steps	Initial Cells	Initial Objects	Number of Rules
Haberman	314	52	36,517	929
Iris	155	49	2.08×10^3	617
Thyroid	223	73	2.76×10^4	1097
Ecoli	344	128	1.18×10^3	2389
Diabetes	778	222	2.76×10^5	6170
Breast	694	235	19,331	6183
Glass	222	132	2.17×10^4	1958
Wine	190	173	1.60×10^5	2340
Vehicle	866	524	1,581,507	15,268
Ionosphere	359	618	2.64×10^3	11,628

Table 3. The accuracy of clusters evaluates the right objects of clusters in each class.

Data Set	The Algorithm	K-Means	AHC	MAFC [27]	Rashno E. et al. [28]
Haberman	47.82%	48.64%	50.06%	-	-
Iris	93.33%	89.79%	91.54%	90.7%	94.66%
Breast	93.99%	96.06%	95.83%	-	91.41%
Wine	97.75%	95.20%	97.73%	-	83.14%
Ionosphere	72.93%	70.20%	70.54%	72.4%	-
Average	81.16 %	79.97%	81.14%	-	-

The intrinsic maximal parallelism of P systems can be exploited to produce a speed-up for solutions. In order to achieve this, the model needs several ingredients, among them the ability to generate an exponential workspace in polynomial time. The computational cost is more than k-means as the last stage of its algorithm is repetitive. Table 4 compares the time consuming against k-means and AHC where the fastest (in average) is shown in boldface. The results show that our algorithm can cluster faster on most data sets.

Table 4. Comparison of time consuming among the three algorithms.

Data Set	The Algorithm	K-Means	AHC
Haberman	0.07 s	0.08 s	0.07 s
Iris	0.03 s	0.05 s	0.08 s
Thyroid	0.05 s	0.04 s	0.06 s
Ecoli	0.07 s	0.04 s	0.09 s
Breast	0.15 s	0.05 s	2.48 s
Glass	0.05 s	0.07 s	0.07 s
Wine	0.04 s	0.07 s	0.05 s
Vehicle	0.19 s	0.14 s	0.46 s
Ionosphere	0.08 s	0.06 s	0.14 s
Average	0.08 s	0.07 s	0.38 s

4. Conclusions

This paper discusses the use of weighted spiking neural P system with anti-spike and astrocyte to appropriately develop a novel hybrid method with grid-density based algorithm for solving clustering problems which first projects the data patterns on a two-dimensional space to overcome the curse of

dimensionality problem. To choose these two well-informed features, a simple and fast feature selection algorithm is proposed. Then, through meshing the plane with grid lines and deleting sparse grids, clusters are found out. In particular, we present weighted spiking neural P systems with anti-spikes and astrocyte (WSNPA2 in short) to implement grid-density based approach in parallel. Each neuron in weighted SN P system contains a spike, which can be expressed by a computable real number. Spikes and anti-spikes are inspired by neurons communicating through excitatory and inhibitory impulses. Astrocytes have excitatory and inhibitory influence on synapses. Characteristic of each dimension is calculated and compared by rules independently in different membranes synchronously. Communications among membranes is utilized to explore clusters. Experimental results on multiple real-world datasets demonstrate the effectiveness and efficiency of our approach to classical k-means, AHC and two other new algorithms.

Author Contributions: Conceptualization, J.X.; methodology, D.K.; formal analysis, Y.W.; writing—original draft, D.K. and X.W.; writing—review and editing, X.L. and J.Q.; supervision, J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 61802234, 61876101), Natural Science Foundation of Shandong Province (No. ZR2019QF007), and China Postdoctoral Project (No. 2017M612339).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, T.; Gong, F.; Liu, X. Spiking Neural P Systems With White Hole Neurons. *IEEE Trans. Nanobiosci.* **2016**, *15*, 666–673. [[CrossRef](#)] [[PubMed](#)]
2. Song, T.; Pan, L.; Wu, T. Spiking Neural P Systems With Learning Functions. *IEEE Trans. Nanobiosci.* **2019**, *18*, 176–190. [[CrossRef](#)] [[PubMed](#)]
3. Wu, T.; Păun, A.; Zhang, G.; Neri, F. Spiking Neural P Systems With Polarizations. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3349–3360. [[PubMed](#)]
4. Song, T.; Zeng, X.; Zheng, P.; Jiang, M.; Rodríguez-Patón, A. A Parallel Workflow Pattern Modeling Using Spiking Neural P Systems With Colored Spikes. *IEEE Trans. Nanobiosci.* **2018**, *17*, 474–484. [[CrossRef](#)]
5. Song, T.; Liu, X.; Zeng, X. Asynchronous Spiking Neural P Systems with Anti-Spikes. *Neural Process. Lett.* **2015**, *42*, 633–647. [[CrossRef](#)]
6. Wang, J.; Hoogeboom, H.; Pan, L.; Păun, G.; Perez-Jimenez, M. Spiking neural P systems with weights. *Neural Comput.* **2010**, *22*, 2615–2646. [[CrossRef](#)]
7. Frias, T.; Diaz, C.; Sanchez, G.; Garcia, G.; Avalos, G.; Perez, H. Four Single Neuron Arithmetic Circuits based on SN P Systems with Dendritic Behavior, Astrocyte-like control and rules on the synapses. *IEEE Lat. Am. Trans.* **2018**, *16*, 38–45. [[CrossRef](#)]
8. Yang, J.; Peng, H.; Luo, X.; Wang, J. Stochastic Numerical P Systems With Application in Data Clustering Problems. *IEEE Access* **2020**, *8*, 31507–31518. [[CrossRef](#)]
9. Zeng, X.; Zhang, X.; Song, T.; Pan, L. Spiking neural P systems with thresholds. *Neural Comput.* **2014**, *26*, 1340–1361. [[CrossRef](#)]
10. Wu, T.; Pan, L.; Yu, Q.; Tan, K.C. Numerical Spiking Neural P Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, 1–15. [[CrossRef](#)]
11. Ma, T.; Hao, S.; Wang, X.; Rodríguez-Patón, A.A.; Wang, S.; Song, T. Double Layers Self-Organized Spiking Neural P Systems With Anti-Spikes for Fingerprint Recognition. *IEEE Access* **2019**, *7*, 177562–177570. [[CrossRef](#)]
12. Song, T.; Pan, L. Spiking Neural P Systems With Rules on Synapses Working in Maximum Spikes Consumption Strategy. *IEEE Trans. Nanobiosci.* **2015**, *14*, 38–44. [[CrossRef](#)] [[PubMed](#)]
13. Cabarle, F.; Adorna, H.; Perez-Jimenez, M.; Song, T. Spiking neural P systems with structural plasticity. *Neural Comput. Appl.* **2015**, *26*, 1905–1917. [[CrossRef](#)]
14. Adl, A.; Badr, A.; Farag, I. Towards a Spiking Neural P Systems OS. *arXiv* **2010**, arXiv:1012.0326.
15. Liu, X.; Li, Z.; Liu, J.; Liu, L.; Zeng, X. Implementation of arithmetic operations with time-free spiking neural P systems. *IEEE Trans. Nanobiosci.* **2015**, *14*, 617–624. [[CrossRef](#)] [[PubMed](#)]

16. Zhang, G.; Rong, H.; Neri, F.; Perez-Jimenez, M. An optimization spiking neural P system for approximately solving combinatorial optimization problems. *Int. J. Neural Syst.* **2014**, *24*, 1440006. [[CrossRef](#)]
17. Păun, G. Computing with Membranes. *J. Comput. Syst. Sci.* **2000**, *61*, 108–143. [[CrossRef](#)]
18. Nishida, T. Membrane algorithm with Brownian sub algorithm and genetic sub algorithm. *Int. J. Found. Comput. Sci.* **2007**, *18*, 1353–1360. [[CrossRef](#)]
19. Zhang, G.; Marian, G.; Wu, C. A quantum-inspired evolutionary algorithm based on P systems for Knapsack problem. *Fundam. Inform.* **2008**, *87*, 93–116.
20. Huang, L.; Suh, I.H.; Abraham, A. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Inf. Sci.* **2011**, *181*, 2370–2391. [[CrossRef](#)]
21. Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the ACM SIGMOD on Management of Data, Seattle, WA, USA, 1–4 June 1998.
22. Zeng, X.; Song, T.; Zhang, X.; Pan, L. Performing Four Basic Arithmetic Operations With Spiking Neural P Systems. *IEEE Trans. Nanosci.* **2012**, *11*, 366–374.
23. Zhao, Y.; Song, J. AGRID: An efficient algorithm for clustering large high dimensional data sets. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Seoul, Korea, 30 April–2 May 2003.
24. Rathore, P.; Kumar, D.; Bezdek, J.C.; Rajasegarar, S.; Palaniswami, M. A Rapid Hybrid Clustering Algorithm for Large Volumes of High Dimensional Data. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 641–654. [[CrossRef](#)]
25. Fahy, C.; Yang, S. Dynamic Feature Selection for Clustering High Dimensional Data Streams. *IEEE Access* **2019**, *7*, 127128–127140. [[CrossRef](#)]
26. Monti, A.; Ponci, F. Power grids of the future: Why smart means complex. In Proceedings of the 2010 Complexity in Engineering, Rome, Italy, 22–24 February 2010; pp. 7–11.
27. Shang, R.; Zhang, W.; Li, F.; Jiao, L.; Stolkin, R. Multi-objective artificial immune algorithm for fuzzy clustering based on multiple kernels. *Swarm Evol. Comput.* **2019**, *50*, 100485. [[CrossRef](#)]
28. Rashno, E.; Minaei-Bidgoli, B.; Guo, Y. An effective clustering method based on data indeterminacy in neutrosophic set domain. *Eng. Appl. Artif. Intell.* **2020**, *89*, 103411. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).