# Data Clustering in Enterprise Computing: A New Generalized Cellular Automata

Dianxun Shuai[1], Qing Shuai[2], Yuming Dong[1], and Liangjun Huang[1]
[1] East China University of Science and Technology,
Shanghai 200237, China, shdx411022@online.sh.cn
[2] Huazhong University of Science and Technology,
Wuhan 430074, China, echoshuai@163.com

**Abstract**. This paper is devoted to novel stochastic generalized cellular automata (GCA) for self-organizing data clustering in enterprise computing.* The GCA transforms the data clustering process into a stochastic process over the configuration space in the GCA array. The proposed approach is characterized by the self-organizing clustering and many advantages in terms of the insensitivity to noise, quality robustness to clustered data, suitability for high-dimensional and massive data sets, the learning ability, and the easier hardware implementation with the VLSI systolic technology. The simulations and comparisons have shown the effectiveness and good performance of the proposed GCA approach to data clustering.

## 1 Introduction

The data clustering in enterprise computing, as a class of data mining technique, is to partition a data set given for enterprise computing into separate clusters, with each cluster being composed of the data objects that possess similar characteristics. Most existing clustering methods can be broadly classified into three categories: partitioning methods, hierarchical methods and locality-based methods [1]-[14].

This paper is devoted to novel generalized cellular automata (GCA) for self-organizing data clustering in enterprise computing. By the GCA clustering methods, the data objects of a given data set in enterprise computing are randomly distributed on a GCA array, and may randomly move back and forth on the GCA array with a probability related to similarity degree. The GCA thus transforms a clustering process into a stochastic self-organizing process over the configuration space of data of other components, leading to standard AND-constraints. However, it might also

* This work was supported by the National Natural Science Foundation of China under Grant No.60473044, No.60575040 and No.60135010.

*Please use the following format when citing this chapter:*

Shuai, D., Shuai, Q., Dong, Y., Huang, L., 2006, in International Federation
for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information
Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.31-39.

objects on the GCA array. The analysis and simulations have revealed very encouraging performance of the GCA approach to data self-organizing clustering in terms of the parallel computation, the insensitivity to noise, the quality robustness to clustered data, the suitability for high-dimensional and massive data sets, the learning ability, the openness and the easier hardware implementation with the VLSI systolic technology.

## 2 Generalized Cellular Automata

The GCA-based self-organizing data clustering in enterprise computing is realized by a two-dimensional $N \times N$ cellular array. At first, all the data objects for enterprise computing are randomly mapped on the GCA array, with the state $s_{ij}(\mathrm{t})$ of the cell $c_{ij}$ being equal to the data object mapped on the cell. If there is no data object mapped on the cell $c_{ij}$, the state $s_{ij}(\mathrm{t})$ is denoted by $\phi$; Then, according to some local dynamic rules, data objects may concurrently and stochastically move back and forth on the GCA array, with a motion probability being determined by a harmony function. Thus the special configuration of data objects on the GCA array forms a Markov stochastic process that may converge to stationary probability distributions. The data object configuration with the maximal stationary probability provides the optimal solution to self-organizing data clustering for enterprise computing.

**Definition 1** *The similarity* $d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))$ *between two cells* $c_{ij}$ *and* $c_{i'j'}$ *at time* $t$ *is defined by*

- *If* $s_{ij}(\mathrm{t}) \neq \phi$ *and* $s_{i'j'}(\mathrm{t}) \neq \phi$ *, then* $0 \leq d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t})) \leq 1$;
- *otherwise,* $d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t})) = -1$.

*The harmony* $\kappa(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))$ *between two cells* $c_{ij}$ *and* $c_{i'j'}$ *at time* $t$ *is defined by*

- If $d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t})) \geq \theta_{\mathrm{ij}}$, then $k(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))\text{=}1$;
- If $0 \leq d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))<\theta_{\mathrm{ij}}$, then $k(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))\text{=-}1$;
- If $d(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t})) = -1$, then $k(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t}))\text{=}0$;

*where* $\theta_{ij}$ *is a positive threshold less than* 1.

*The harmony* $h_{ij}(\mathrm{t})$ *of the cell* $c_{ij}$ *at time* $t$ *is defined by*

$$h_{ij}(\mathrm{t})\text{=} \sum_{(i',j')\in N_{ij}} k(s_{ij}(\mathrm{t}),\ s_{i'j'}(\mathrm{t})). \tag{1}$$

*where* $N_{ij} = \left\{c_{i,j\text{-}1}, c_{i,j+1}, c_{i-1,j}, c_{i+1,j}\right\}$ *is the neighbor of the cell* $c_{ij}$.

**Definition 2** *The matrix* $\Gamma(t) = \left[s_{ij}(t)\right]_{N\times N}$ *represents a special distribution of data objects on the GCA array at time* $t$. *The aggregate harmony of* $\Gamma(t)$ *is data*

*data objects on the GCA array at time $t$. The aggregate harmony of $\Gamma(t)$ is defined by*

$$\mathrm{H}(\Gamma(t)) = \sum_{i,j} \omega_{ij} h_{ij}(t), \tag{2}$$

*where $\omega_{ij}$ is a weight coefficient which may be obtained by using a learning algorithm from the given priori probability distribution of data objects.*

**Definition 3** *A local dynamic rule for every cell in the GPM array is described in the format:*

$$[\textit{cell state } S \textit{ at time } t] \xrightarrow[\textit{action}\,\mathrm{A}]{\textit{input}\,\mathrm{I}} [\textit{cell state } S' \textit{ at time}(t+1)] / \textit{probalitity} P ,$$

*which implies that the input* I *imposed to a cell at time t will, with the probability $P$, trigger the action* A *and the state transition from $S$ to $S'$. For any cell $c_{ij}$ in a GCA array, i, j =* 1, 2,...,*N, the local dynamic rule set* R *is composed of the following six rules :*

$$\mathrm{R}(1): [s_{ij}(t) = \phi] \xrightarrow[\textit{receive a data object } \kappa]{c_{ij} \textit{ is selected initially}} [s_{ij}(t+1) = \kappa] / p = 1;$$

$$\mathrm{R}(2): [s_{ij}(t) \neq \phi] \xrightarrow[\textit{pass } s_{ij}(t) \textit{ to } c_{i'j'}]{c_{ij} \textit{ selects } c_{i'j'}} [s_{ij}(t+1) = \phi] / p = f(\Delta\mathrm{H});$$

$$\mathrm{R}(3): [s_{ij}(t) \neq \phi] \xrightarrow[\textit{not pass } s_{ij}(t) \textit{ to } c_{i'j'}]{c_{ij} \textit{ selects } c_{i'j'}} [s_{ij}(t+1) = s_{ij}(t)] / p = 1 - f(\Delta\mathrm{H});$$

$$\mathrm{R}(4): [s_{ij}(t) = \phi] \xrightarrow[\textit{receive } s_{i'j'}(t)]{c_{ij} \textit{ is selected by } c_{i'j'}} [s_{ij}(t+1) = s_{i'j'}(t)] / p = f(\Delta\mathrm{H});$$

$$\mathrm{R}(5): [s_{ij}(t) = \phi] \xrightarrow[\textit{no action taken}]{c_{ij} \textit{ is not selected}} [s_{ij}(t+1) = \phi] / p = 1;$$

$$\mathrm{R}(6): [s_{ij}(t) \neq \phi] \xrightarrow[\textit{no action taken}]{c_{ij} \textit{ is not selected}} [s_{ij}(t+1) = s_{ij}(t)] / p = 1,$$

*where the function $f(\square)$ is defined by* $f(\Delta\mathrm{H}) = \dfrac{1}{1+e^{\Delta\mathrm{H}/T}}$ (3)

$\Delta\mathrm{H} = \mathrm{H}(\Gamma(t)) - \mathrm{H}(\Gamma(t+1))$ *is the harmony increment of the GCA array if the given event indicated by the local dynamic rule really happens; The temperature T is a parameter used to improve the solution quality.*

The above local transitive rules can also be illustrated in Fig. 1.

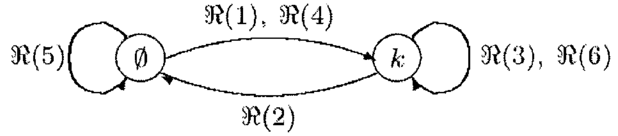


**Fig. 1.** The transition diagram of the cellular state of a cell by using the local dynamic rule set R , where the symbol $k$ represents a data object mapped on a cell.

**Remarks 1:**

- **The motivation for cell harmony** : By the Eq. (1), the larger the total

- similarity between the cell $c_{ij}$ and all the cells in its neighbor $N_{ij}$ , the larger the harmony $h\left(c_{ij}(t)\right)$ is. Moreover if there is such a cell in $N_{ij}$ that holds no data object, then the harmony 0 between the cell and the cell $c_{ij}$ has no affect on the harmony of the cell $c_{ij}$.

- **The motivation for GCA harmony :** The aggregate harmony of a configuration of data objects on the GCA array may describe the global average similarity degree among all the cells and their neighboring cells. Intuitively it is expected that a better clustering should correspond to a data object configuration with larger aggregate harmony. Therefore for two configurations, *u* and *v*, of data objects on the GCA array if the aggregate harmony $\mathrm{H}(u)$ is larger than the aggregate harmony $\mathrm{H}(v)$ ; intuitively the probability of the transition from *u* to *v* should be less than the probability from *v* to *u*. The Eq. (3) is consistent with this intuition.

## 3 GCA Parallel Algorithm and Properties

To implement the GCA for self-organizing data clustering in enterprise computing, the algorithm GCAA uses two independent cellular array I and cellular array II to store two different configurations of data objects on the GCA array.

**The Parallel Algorithm GCAA :**

**Costep 1.** The state of every cell in the array I and array II is set to $\phi$, indicating no data object mapped on any cell.

**Costep 2.** All the data objects in the given data set are concurrently and randomly mapped on the array I and array II, forming two different configurations of data objects on them.

**Costep 3.** Every cell $c_{ij}$ in the array I and array II concurrently compute its harmony $h_{ij}(t)$ by Eq. (1).

**Costep 4.** The array I and array II concurrently compute their aggregate harmonies by Eq.(2).

**Costep 5.** Compute the transitive probability by Eq.(3), and then, upon the transitive probability, randomly choose one configuration from the array I and array II.

**Costep 6.** Using a random configuration, update the configuration on the array I or array II which has not been chosen in Costep 5.

**Costep 7.** Repeat Costep 3 through Costep 6 until reaching a stationary probability distribution over the configuration space on the array I and array II, where the configuration with the maximal probability corresponds to the optimal clustering.

We summarize the properties of the algorithm GCAA by the following Lemmas and Theorems whose proofs are given in Appendix.

**Lemma 1** *Carrying out the algorithm GCAA is equivalent to carrying out the* *dynamic rules* R .

**Lemma 2** *The stochastic process* $\{\Gamma(t), t = 0, 1, \cdots\}$ *generated by executing the algorithm GCAA is a finite homogeneous Markov chain.*

**Lemma 3** *The stochastic process* $\{\Gamma(t), t = 0, 1, \cdots\}$ *generated by the algorithm GCAA is an irreducible homogeneous Markov chain, where all the configurations are positive recurrent, that is, any configuration may return to itself in a finite time period with the probability 1.*

**Lemma 4** *The stochastic process* $\{\Gamma(t), t = 0, 1, \cdots\}$ *generated by the algorithm GCAA must reach a stationary probability distribution over the configuration space, that is, any configuration may occur finally with a fixed probability. Moreover the stationary distribution is independent of the initial configuration.*

**number of executions of GCAA:1000**

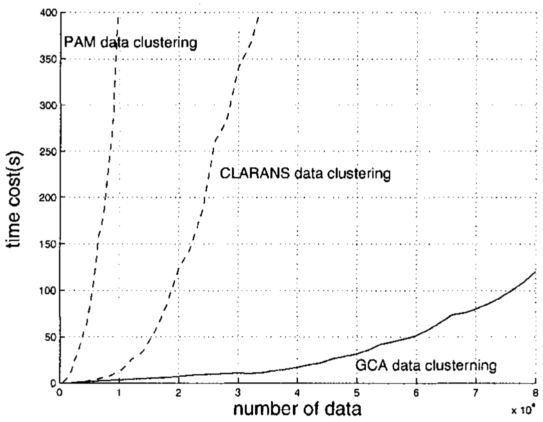


**Fig. 2.** The comparisons of clustering time between the proposed GCAA algorithm and PAM, CLARANS algorithms, where the best of 5 local optimal solutions for CLARANS is shown. The number of cluster: 20; The size of the data set: from 2000 to 80000. The algorithm GCAA has better performance.

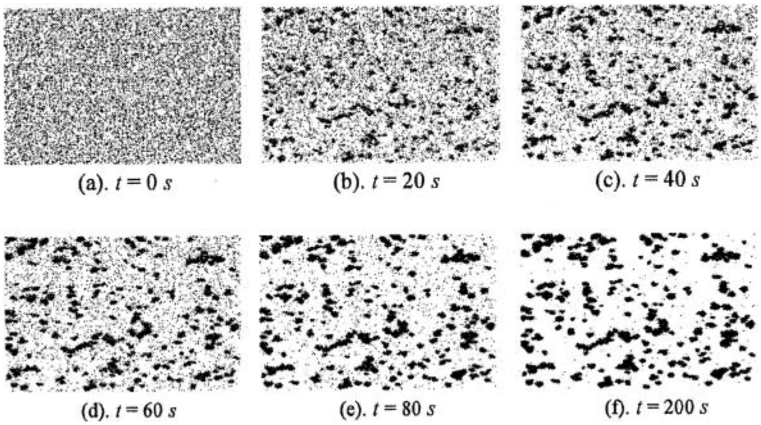
**Theorem 1** *The stationary probability distribution obtained by executing the algorithm GCAA is the maximal entropy distribution over the configuration space. Moreover the stochastic configuration with the maximal probability has the maximal aggregate harmony.*

**Theorem 2** *A configuration with the maximal probability in a stationary probability distribution produced by the algorithm GCAA must be such a special configuration of data objects on the GCA array that has the minimal number of connected regions, with each region corresponding to a distinct cluster of the given data set.*

## 4 Simulations and Comparisons

Using the GCAA algorithm, the simulations on different data set sizes, different data types, various cluster shapes, noisy data set and higher dimensional data are carried out respectively.

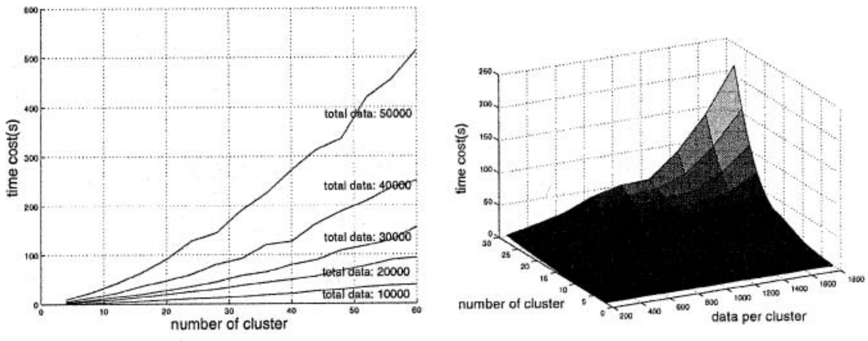
The comparison of clustering time between the proposed GCAA algorithm and PAM, CLARANS algorithms is given in Fig. 2. We have also made many experiments and comparisons on the ability to eliminate outliers and to deal with high-dimensional data, details omitted here for the page limitation.



(a). $t = 0$ $s$ (b). $t = 20$ $s$ (c). $t = 40$ $s$

(d). $t = 60$ $s$ (e). $t = 80$ $s$ (f). $t = 200$ $s$

**Fig. 3.** The transient configurations of data objects on a GCA array during executing the algorithm GCAA, where $t$ is the number of iterations. Number of clusters : 60; Data set size : 20000.



(a) (b)

**Fig. 4.** The relation of GCA-based clustering time with the number of clusters and the data set size. (a). The relationship between the clustering time and the number of clusters in a data set, where number of clusters: 4-60; data set size: 10000-50000; (b). The clustering time changes with the data set size and number of clusters, where number of clusters: 2-30; data per cluster: 200-1800.

Several snaps of data clustering on the GCA array during executing the algorithm GCAA are illustrated in Fig. 3, where the initial configuration of data objects randomly mapped on the GCA array is demonstrated in Fig. 3(a), and finally almost all the similar data are successfully clustered together as shown in Fig. 3(f). The relation between the GCA clustering time, the number of clusters and the data set size is shown in Fig. 4.

## 5 Conclusions

The GCA approach and algorithm GCAA for self-organizing data clustering in enterprise computing have been proposed in this paper. The analysis and simulations on a variety of data sets given for enterprise computing have shown many advantages over other widely used clustering algorithms in terms of the followings:

- Faster clustering speed for both the large data set and the noisy data set;
- The ability to handle and recognize the shape-varying and size-varying clusters; The robustness to outliers, and the ability to eliminate the noise data from clustering result;
- The ability to learn the priori probability distribution of data objects and to use the previous clustering results for shortening the clustering time and improving the clustering quality;
- The suitability for high dimensional data sets, the clustering performance being not obviously affected by the data dimension.

## References

1. T. Thanh, N. Wehrens, R. Buydens, and M. C. Lutgarde, A Clustering Algorithm For Multispectral Images, *Analytica Chimica Acta* **490**, 303-312 (2003).

2. L. Kaufman, and P. J. Rousseeuw, Finding Groups in Data, *Introduction to Cluster Analysis*, (Addison Wesley, NEW York, 1990).

3. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum Likelihood from Incomplete Data Via the EM Algorithm, *Journal of the Royal Statistical Society Series B* **39**(1), 1-38 (1977).

4. J. Han, and M. Kamber, Data Mining: Concepts and Techniques, (Morgan Kaufmann, 2000).

5. G. Karypis, E. H. Han, and V. Kumar. CHAMELEON. A Hierarchical Clustering Algorithm Using Dynamic Modelling, *Computer* **32**, 68-75 (1999).

6. S. Guha, R. Rastogi, and K. Shim, Rock: A Robust Clustering Algorithm for Categorical Attributes, In Proceedings of the 1999 International Conference on Data Engineering, Sydney, Australia, March 1999, 512-521 (1999).

7. R. Kannan, S.Vempala, and A.Vetta, On Clustering: Good, Bad, and Spectral, In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, (1999).

8. D. Cheng, R. Kannan, S. Vempala, and G. Wang, On a Recursive Spectral Algorithm for Clustering from Pairwise Similarities, MIT LCS Technical Report, MIT-LCS-TR-906, 2003.

9. J. Shi and J. Malik, Normalized Cuts And Image Segmentation, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **22**, 887-905 (2000).

10. H. Frigui, An Efficient Clustering Approach To Identify Clusters of Arbitrary Shapes in Large Data Sets, In Proceedings of the ACM SIGKDD, Conference Knowledge Discovery and Data Mining, 507-512 (2002).

11. K. Szczubialka, J. Verdu-Andres, and D.L. Massart, A New Method of Detecting Clustering in the Data, *Chemometrics and Intelligent Laboratory Systems* **41**, 145-160 (1998).

12. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, In Proceedings of the ACM SIGMOD, (1999).

13. H. Frigui and M. Rhouma, A Synchronization Based Algorithm for Discovering Ellipsoidal Clusters in Large Data-Sets, In Proceedings of the IEEE Conference on Data Mining, (2001).

14. L.O. Chua, and L. Yang, Cellular Neural Networks: Theory, *IEEE Transaction on Circuits and Systems* **35**(10), 1257-1272 (1988).

15. P. S. Shelokar, V. K. Jayaraman, and B.D. Kulkarni, An Ant Colony Approach for Clustering, *Analytica Chimica Acta* **509**(2), 187-196 (2004).

## Appendix

**Proof of Lemma 1.** Randomly preparing two configurations in the array I and array II in the Costep 2 and Costep 6 amounts to alternately using the two arrays to obtain a configuration at the current time and a configuration at the next time in the same time session $\tau$ . Using the $f\left(\Delta \mathrm{H}\right)$ to randomly choose a configuration from the array I and II in the Costep 5 is equivalent to carrying out the rule $\mathrm{R}$ simultaneously on all the cells, with the probability $f\left(\Delta \mathrm{H}\right)$. As a result, Carrying out the algorithm GCAA is equivalent to carrying out the transitive rule $\mathrm{R}$ .

**Proof of Lemma 2.** The number of the possible configurations for the stochastic process $\left\{\Gamma\left(t\right), t=0,1,\cdots\right\}$ is finite and equal to $\binom{N^2}{n}\left(n!\right)$. The transition probability from $\Gamma\left(t\right)$ to $\Gamma\left(t+1\right)$ only depends upon

$\Delta \mathrm{H}=\mathrm{H}\left(\Gamma\left(t\right)\right)-\mathrm{H}\left(\Gamma\left(t+1\right)\right)$ and is independent of the configuration $\Gamma\left(t-i\right), i\geq 1$, and independent of the time $t$.

**Proof of Lemma 3.** By Eq.3, any two stochastic configurations are accessible from each other in the sense that the transitive probability from one to another is larger than zero. Thus the unique closed set in the configuration space is the set that is composed of all the possible configurations, which implies that it is irreducible. Then it follows from Lemma 2 that all the configurations of a finite inter-accessible Markov chain are all positive recurrent.

**Proof of Lemma 4.** By the Markov process theory, if any two stochastic states in the state space may transit by one-step from one to another with a non-zero probability, then the stochastic process must converge to a stationary state. Since any two configurations in the configuration space have finite aggregate harmonies by Eq.(1) and Eq.(2), and the transitive probability from one to another is larger than zero by the Eq.(3) accordingly, it turns out that the stochastic configuration sequence $\left\{\Gamma\left(t\right), t=0,1,\cdots\right\}$ must finally reach a stationary probability distribution.

**Proof of Theorem 1.** Number all the data objects that have been mapped onto the GCA array by 1 through n, and assume that, in the given stochastic configuration $u\in\Gamma^*$ , the k-th data object is carried by the cell $c_{i(u,k),j(u,k)}$ in the GCA array , $i\left(u,k\right), j\left(u,k\right)\in\left\{1,\cdots,N\right\}$ , with its harmonic function value denoted by $h_{i(u,k),j(u,k)}$ . At first, we try to obtain a stationary probability distribution $p_u, u \in I^*$, such that it has the maximal entropy $\max_{p_u}\left[-\sum_{u\in I^*} p_u \ln p_u\right]$, and satisfies the following constraints:

$$\sum_{u\in I^*} p_u = 1, and \sum_{u\in I^*} p_u h_{i(u,k),j(u,k)} = \overline{h_k}\ for\ \forall k \in \{1,\cdots,n\}. \qquad (5)$$

where $\overline{h_k}$ is constant related to $k$, upon a priori probability with respect to k-th data object in data sets.

$$From\ \frac{\partial}{\partial p_u}\left\{\sum_{u\in I^*} p_u \ln p_u - \sum_k\left[\sum_{u\in I^*} h_{i(u,k),j(u,k)} - \overline{h_k}\right] - \lambda\left[\sum_{u\in I^*} p_u \ln p_u - 1\right]\right\} = 0,$$

we obtain $p_u^* = Z^{-1}\left\{\exp\sum_{k=1}^{n} h_{i(u,k),j(u,k)}\right\} = Z^{-1}\exp(\mathrm{H}(u)),$

where $Z = \sum_{u\in I^*}\exp\left\{\sum_k h_{i(u,k),j(u,k)}\right\}$; $\mathrm{H}(u)$ is the harmonic function value of the stochastic configuration u.

On the other hand, we make use of $\pi^*$ and $\pi_u^*$ to denote the stationary probability distribution and the stationary probability of the configuration u, respectively, which is yielded by concurrently executing the rule R . Then

$$\frac{\pi_u^*}{\pi_v^*} = \frac{e^{\mathrm{H}(u)/T}}{e^{\mathrm{H}(v)/T}}\ and\ \pi_u^* = Z^{-1}e^{\mathrm{H}(u)/T}$$

can be derived from $f(\mathrm{H}_t(v) - \mathrm{H}_{t+1}(u))\pi_v^* = f(\mathrm{H}_t(u) - \mathrm{H}_{t+1}(v))\pi_u^*$, and $\mathrm{H}_t(v) - \mathrm{H}_{t+1}(u) = -(\mathrm{H}_t(u) - \mathrm{H}_{t+1}(v))$.

It therefore follows that $\pi_u^* = p_u^*$ upon the temperature T, which implies that the stationary probability distribution of the stochastic process decided by the GCAA is just the maximal entropy distribution.

Moreover, it turns out that the stochastic configuration with the maximal harmonic function value has the maximal probability over the obtained stationary probability distribution.

**Proof of Theorem 2.** By contrary, assume that, when the dynamics of GCA has reached to the stationary probability distribution, the configuration *u* with the maximal harmony function makes some data objects that should be partitioned into the same cluster be distributed in more than one unconnected region in the GCA array. It follows from Eq. (2) that laying the same data objects in several separate unconnected regions of cells gives rise to smaller harmony function of these cells than laying in one connected region of cells. It turns out that the configuration *u* does not have the maximal value of harmony function, resulting in the contradiction to the assumption. By Theorem 1, therefore, the configuration with the maximal probability in the stationary probability distribution has the minimal number of connected regions in GCA array.