Supplementary Material

A shrinking synchronization clustering algorithm based on a linear weighted Vicsek model

Xinquan Chen[a,b,*], Jianbo Ma[c] , Yirou Qiu[d] , Sanming Liu[a] , Xiaofeng Xu[a] and Xianglin Bao[a]
[a]*Industrial Innovation Technology Research Co. Ltd., Anhui Polytechnic University, Wuhu, 241000, China*
[b]*School of Computing, Macquarie University, Sydney, NSW, 2109, Australia*
[c]*Dolby Laboratories, Sydney, NSW, 2060, Australia*
[d]*Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, N2L3G1, Canada*

Online Resource 1. Other basic definitons and properties

Sdefinition 1. The δ near neighbor point set $N_\delta(\boldsymbol{x})$ of point $\boldsymbol{x}$ is defined as:

$$N_\delta(\boldsymbol{x}) = \{\boldsymbol{y} \mid 0 < \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) \le \delta, \boldsymbol{y} \ne \boldsymbol{x}, \boldsymbol{y} \in D\}, \tag{s1}$$

where $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})$ is the dissimilarity measure between point $\boldsymbol{x}$ and point $\boldsymbol{y}$ in the data set $D = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$. Parameter δ is a predefined threshold.

Sdefinition 2 (Böhm et al., 2010)**.** Point $\boldsymbol{x} = (x_1, \ldots, x_d)$ is a vector in d-dimensional Euclidean space. If each point $\boldsymbol{x}$ is regarded as a phase oscillator based on Kuramoto model, with an interaction in the δ near neighbor point set $N_\delta(\boldsymbol{x})$, then the dynamics of the k-th dimension x_k ($k = 1, 2, \ldots, d$) of point $\boldsymbol{x}$ over time is described by:

$$x_k(t+1) = x_k(t) + \frac{1}{|N_\delta(\boldsymbol{x}(t))|} \sum_{\boldsymbol{y} \in N_\delta(\boldsymbol{x}(t))} \sin\left(y_k(t) - x_k(t)\right), \tag{s2}$$

where $\boldsymbol{x}(t = 0) = (x_1(0), \cdots, x_d(0))$ represents the original phase of point $\boldsymbol{x}$, $x_k(t+1)$ describes the renewal phase value in the k-th dimension of point $\boldsymbol{x}$ at the t-step evolution, and $\boldsymbol{y} = (y_1, \ldots, y_d)$ is a δ near neighbor point of point $\boldsymbol{x}$ at the t-step evolution.

Sdefinition 3 (Chen, 2017). The t-step δ near neighbor undirected graph $G_\delta(t)$ of the data set $D = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ is defined as:

$$G_\delta(t) = (V(t), E(t)), \tag{s3}$$

where $V(t = 0) = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ is the original vertex set, $E(t = 0) = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_j \in N_\delta(\boldsymbol{x}_i), \boldsymbol{x}_i$ $(i = 1, \ldots, n) \in D\}$ is the original edge set. $V(t) = \{\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_n(t)\}$ is the t-step vertex set of the data set D, $E(t) = \{(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) \mid \boldsymbol{x}_j(t) \in N_\delta(\boldsymbol{x}_i(t)), \boldsymbol{x}_i(t)$ $(i = 1, \ldots, n) \in V(t)\}$ is the t-step edge set, and the weight computing equation of edge $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is $weight(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Sdefinition 4 (Chen, 2017). The t-step average length of edges, AveLen(t), in a t-step δ near neighbor undirected graph $G_\delta(t)$ is defined as:

*Corresponding author. E-mail: chenxqscut@126.com.

$$\text{AveLen}(t) = \frac{1}{|E(t)|}\sum_{e\in E(t)}|e|, \tag{s4}$$

where $E(t)$ is the t-step edge set of $G_\delta(t)$, and $|e|$ is the length (or weight) of edge e. The average length of edges in $G_\delta(t)$ decreases to its limit 0, that is $\text{AveLen}(t) \to 0$, as more δ near neighbor points synchronize together with time evolution. In our algorithm, AveLen(t) can be used to characterize the degree of local synchronization.

Sdefinition 5 (Böhm et al., 2010). The cluster order parameter r_c characterizing the degree of local synchronization is defined as:

$$r_c = \frac{1}{n}\sum_{i=1}^{n}\sum_{\boldsymbol{y}\in N_\delta(\boldsymbol{x}_i)} e^{-\text{dist}(\boldsymbol{x}_i,\boldsymbol{y})} \tag{s5}$$

Sdefinition 6 (Chen, 2017)**.** A linear version of Vicsek model for clustering is defined as:

Point $\boldsymbol{x} = (x_1, \cdots, x_d)$ is a vector in d-dimensional Euclidean space. If each point $\boldsymbol{x}$ is regarded as an agent according to a linear version of Vicsek model, with an interaction in the δ near neighbor point set $N_\delta(\boldsymbol{x})$, then the dynamics of point $\boldsymbol{x}$ over time according to Jadbabaie et al. (2003) and Wang et al. (2009) is described by:

$$\boldsymbol{x}(t+1) = \frac{1}{1+|N_\delta(\boldsymbol{x}(t))|}\left(\boldsymbol{x}(t) + \sum_{\boldsymbol{y}\in N_\delta(\boldsymbol{x}(t))}\boldsymbol{y}\right), \tag{s6}$$

where $\boldsymbol{x}(t=0) = (x_1(0), \cdots, x_d(0))$ represents the original location of point $\boldsymbol{x}$, and $\boldsymbol{x}(t+1)$ describes the renewal location of point $\boldsymbol{x}$ at the t-step evolution.

The definitions of three information-theoretic measures:

Sdefinition 7. Mutual Information (MI): the mutual information of two discrete random variables X and Y is defined as:

$$MI(X,Y) = \sum_{x\in X}\sum_{y\in Y}\left(p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\right), \tag{s7}$$

where $p(x, y)$ is the joint probability distribution function of two random variables X and Y, and $p(x)$ is the marginal probability distribution function of random variable X. So as $p(y)$.

Sdefinition 8. Normalized Mutual Information (**NMI**, (Strehl et al., 2002)): the normalized mutual information of two clustering results X and Y is defined as:

$$NMI(X,Y) = \frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}, \tag{s8}$$

where $MI(X, Y)$ is the mutual information of two clustering results X and Y, $H(X)$ and $H(Y)$ are the entropies associated with the clustering results X and Y respectively.

Sdefinition 9. Adjusted Mutual Information (**AMI**, (Vinh et al., 2010)): the adjusted mutual information of two clustering results X and Y is defined as:

$$AMI(X,Y) = \frac{MI(X,Y) - E\{MI(X,Y)\}}{\max\{H(X),H(Y)\} - E\{MI(X,Y)\}}, \tag{s9}$$

where $MI(X, Y)$ is the mutual information of two clustering results X and Y, $H(X)$ and

$H(Y)$ are the entropies associated with two clustering results X and Y respectively, and $\mathrm{E}\{MI(X, Y)\}$ is the expected mutual information of two clustering results X and Y.

Online Resource 2. The descriptions of SynC algorithm and ESynC algorithm

OR2.1 The description of SynC algorithm

The original synchronization clustering algorithm named as SynC is developed by Böhm et al. (Böhm et al., 2010). In order to make a difference between SynC algorithm and our algorithm, we introduce it below using our language according to the description of (Böhm et al., 2010).

1. The description of SynC algorithm

Stable 1. The main procedure of SynC algorithm.

Algorithm Name: The original Synchronization Clustering algorithm (SynC; Böhm et al., 2010).

Input: Dataset $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, dissimilarity measure dist$(\cdot, \cdot)$ and range parameter δ;
Output: The final convergent result $D(T) = \{\boldsymbol{x}_1(T), \ldots, \boldsymbol{x}_n(T)\}$ of the original dataset D;
Procedure: function SynC (D, δ)
/* Initialization: */
1: IterativeStep t is set as zero firstly, that is: $t \leftarrow 0$;
2: **for** $i = 1, 2, \ldots, n$ **do**
3: $\boldsymbol{x}_i(t) \leftarrow \boldsymbol{x}_i$;
4: **end for**
/* Execute the iterative synchronization process of the dynamical clustering: */
5: **while** ((the dynamical clustering does not satisfy its convergent condition) **and** ($t < 20$)) **do**
6: **for** $i = 1, 2, \ldots, n$ **do**
7: Construct the δ near neighbor point set $N_\delta(\boldsymbol{x}_i(t))$ for each point $\boldsymbol{x}_i(t)$ ($i = 1, 2, \ldots, n$) by using Eq.(s1) of Sdefinition 1 of Online Resource 1 of Supplementary Material;
8: Compute the renewal value, $\boldsymbol{x}_i(t+1)$, of $\boldsymbol{x}_i(t)$ by using Eq.(s2) of Sdefinition 2 of Online Resource 1 of Supplementary Material;
9: **end for**
10: Compute the cluster order parameter of all points, r_c, using Eq.(s5) of Sdefinition 5 of Online Resource 1 of Supplementary Material;
11: IterativeStep t is increased by 1, that is: t++;
12: **if** (r_c converges or (t == 20)) **then**
13: We think the dynamical clustering reaches its convergent result, and then exit from the while repetition;
14: **end if**
15: **end while**
16: Finally, we get a convergent result $D(T) = \{\boldsymbol{x}_1(T), \ldots, \boldsymbol{x}_n(T)\}$, where T is the times of the while repetition from step 5 to step 15. The final convergent set $D(T)$ reflects the natural clusters or isolates of the dataset D.

OR2.2 The description of ESynC algorithm

Effective Synchronization Clustering algorithm (ESynC) is developed by Chen (Chen, 2017). In order to make a difference between ESynC algorithm and our algorithm, we introduce it simply below.

Stable 2. The main procedure of ESynC algorithm.

Algorithm Name: An Effective Synchronization Clustering algorithm (ESynC; Chen, 2017).

nput: Dataset $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, dissimilarity measure dist$(\cdot, \cdot)$ and range parameter δ;
Output: The final convergent result $D(T) = \{\boldsymbol{x}_1(T), \ldots, \boldsymbol{x}_n(T)\}$ of the original dataset D;
Procedure: function ESynC (D, δ)
/* Initialization: */
1: IterativeStep t is set as zero firstly, that is: $t \leftarrow 0$;
2: **for** $i = 1, 2, \ldots, n$ **do**
3: $\boldsymbol{x}_i(t) \leftarrow \boldsymbol{x}_i$;
4: **end for**
/* Execute the iterative synchronization process of the dynamical clustering: */
5: **while** ((the dynamical clustering does not satisfy its convergent condition) **and** ($t < 20$)) **do**
6: **for** $i = 1, 2, \ldots, n$ **do**
7: Construct the δ near neighbor point set $N_\delta(\boldsymbol{x}_i(t))$ for each point $\boldsymbol{x}_i(t)$ ($i = 1, 2, \ldots, n$) by using Eq.(s1) of Sdefinition 1 of Online Resource 1 of Supplementary Material;
8: Compute the renewal value, $\boldsymbol{x}_i(t+1)$, of $\boldsymbol{x}_i(t)$ by using Eq.(s6) of Sdefinition 2 of Online Resource 1 of Supplementary Material;
9: **end for**
10: Compute the t-step average length of edges of all points, Ave_len(t), by using Eq.(s4) of Sdefinition 4 of Online Resource 1 of Supplementary Material;
11: IterativeStep t is increased by 1, that is: t++;
12: **if** (Ave_len(t) $\rightarrow$ 0) **then**
13: The dynamical clustering reaches its convergent result, and then exit from the while repetition;
14: **end if**
15: **end while**
16: Finally, we get a convergent result $D(T) = \{\boldsymbol{x}_1(T), \ldots, \boldsymbol{x}_n(T)\}$, where T is the times of the while repetition from step 5 to step 15. The final convergent set $D(T)$ reflects the natural clusters or isolates of the dataset D.

Online Resource 3. The description of experimental data sets

Stable 3. The description of experimental data sets

(a) The description of some kinds of artificial data sets

Data Sets (*DS*)	**Predefined Number of Clusters (*NC*)**	**With Noise**	**Cluster Semidiameter (*CS*)**	**Dimension (*d*)**
DataType1	7	1 isolate	-	2
DataType2	9	no	30	2
DS1	5	yes	40	2
DS2	5	no	50	2
DS3	9	yes	30	2
DS4	9	no	40	2
DS5	12	no	30	2
DS6	12	no	30	4
DS7	12	no	30	6
DS8	12	no	30	8
DS9	5	no	30	2
DS10	5	no	30	4
DS11	5	no	30	6
DS12	5	no	30	8
DS13	5	no	30	20
DS14	5	no	30	40
DS15	5	no	30	80
DS16	5	no	30	100

(b) The description of several UCI data sets

UCI Data Sets	**Number of Points (*n*)**	**Dimension (*d*)**	**Number of Classes (*c*)**
Iris	150	4	3
Wine	178	13	3
Wdbc	569	30	2
Glass	214	9	6
Ionosphere	351	34	2
Letter-recognition	20000	16	26
Segmentation	210	19	7
Cloud	2048	10	2

UCI Data Sets [55]	# Instances	# Attributes	# Classes	The distribution of Classes	Detailed Informaton
Iris	150	4	3	{Setosa: 50, Versicolor: 50, Virginica: 50}	A small classic dataset from Fisher, 1936. One of the earliest datasets used for evaluation of classification methodologies.
Wine	178	13	3	{1: 59, 2: 71, 3: 48}	Using chemical analysis determine the origin of wines.
Wdbc	569	30	2	{B: 357, M: 212}	-
Glass	214	9	6	{Window: {FB: 70, FV: 17, NFB: 76}, Non-window: {C: 13, T: 9, H: 29}}	From USA Forensic Science Service, defined in terms of their oxide content.
Ionosphere	351	34	2	{Good: 225, Bad: 126}	Classification of radar returns from the

					ionosphere.
Letter-recognition	20000	16	26	{A: 443, B: 460, C: 449, …, Z: 408}	Database of character image features; try to identify the letter.
Segmentation	210	19	7	{Brickface: 30, Sky: 30, Foliage: 30, Cement: 30, Window: 30, Path: 30, Grass: 30}	-
Cloud	2048	10	2	{1: 1024, 2: 1024}	Little Documentation

(c) The description of three bmp picture data sets (obtained from Internet)

Picture Data Sets	**Number of Pixels (*n*)**	**Dimension (*d*)**
Picture1	100*100	3
Picture2	100*100	3
Picture3	200*200	3

R3.1 DataType1 used in Fig. 1 is created by Python language

center = [[-200, 100],[-100, -100],[-200, -100], [0, 0], [100, 100],[100, 200], [200, 200]]

cluststd = [25, 20, 25, 15, 22, 25, 20]

x, y = make_blobs(n_samples=1000, n_features=2, centers = center, cluster_std = cluststd, shuffle = False, random_state = 1)

OR3.2 Other artificial data sets (DataType2, DS1 - DS16) used in simulated experiments are created by two C functions

```
#define DIMENSION 2
#define NUM_POINT 300
#define NUM_CLUSTER 5
#define MIN_DISTANCE 18
#define WIDTH 600
#define CLUSTER_DISTANCE 40

int Create_DataSet(struct Point DS[],    int DataClass[], int NumPoint, int NumCluster)
{
      int i, j, k;
      struct Point *Core;    /* Array Core stores the top left corners of clusters. */
      Core = (struct Point *) malloc(sizeof(struct Point) * NumCluster);
      if(Core == NULL)
      {
            printf("There is no enough space for Core[]\n");
            return -1;
      }
      for(i = 0; i < NumCluster; i++)
      {
            for(k = 0; k < DIMENSION; k++)
            {
                  Core[i].x[k] = CLUSTER_DISTANCE + (double) rand() / ((double) RAND_MAX + 1.0) * WIDTH;
            }
      }
      int l = 0;
      for(i = 0; i < NumCluster; i++)
      {
            for(j = 0; j < NumPoint / NumCluster -2; j++)
            {
                  for(k = 0; k < DIMENSION; k++)
                  {
                        DS[l].x[k] = Core[i].x[k] + (double) rand() / ((double) RAND_MAX + 1.0) * CLUSTER_DISTANCE;
                  }
                  DataClass[l] = i;      /* The class number of the l-th data point is assigned by the i-th cluster. */
                  l++;
            }
      }
      while(l < NumPoint)
      {
            for(k = 0; k < DIMENSION; k++)
            {
                  DS[l].x[k] = CLUSTER_DISTANCE + (double) rand() / ((double) RAND_MAX + 1.0) * WIDTH;
            }
```

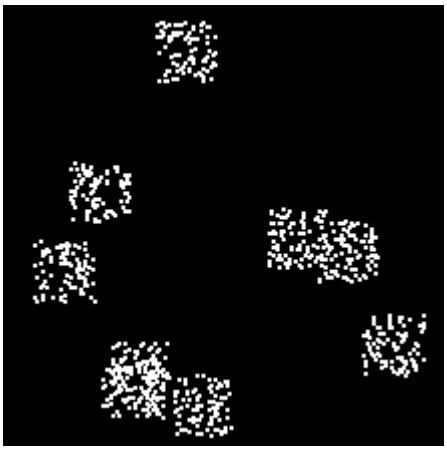
```
            DataClass[l] = -1;     /* The l-th point may be a noise. */
            l++;
        }
        if(Core != NULL)
        {
            free(Core);
            Core = NULL;
        }

        return 0;
}

int Create_DataSetNoNoise(struct Point DS[],    int DataClass[], int NumPoint, int NumCluster)
{
        int i, j, k;
        struct Point *Core;
        Core = (struct Point *) malloc(sizeof(struct Point) * NumCluster);
        if(Core == NULL)
        {
            printf("There is no enough space for Core[]\n");
            return -1;
        }
        for(i = 0; i < NumCluster; i++)
        {
            for(k = 0; k < DIMENSION; k++)
            {
                Core[i].x[k] = CLUSTER_DISTANCE + (double) rand() / ((double) RAND_MAX + 1.0)
* WIDTH;
            }
        }
        int l = 0;
        for(i = 0; i < NumCluster; i++)
        {
            for(j = 0; j < NumPoint / NumCluster -2; j++)
            {
                for(k = 0; k < DIMENSION; k++)
                {
                    DS[l].x[k] = Core[i].x[k] + (double) rand() / ((double) RAND_MAX + 1.0) *
CLUSTER_DISTANCE;
                }
                DataClass[l] = i;
                l++;
            }
        }
        while(l < NumPoint)
        {
            for(k = 0; k < DIMENSION; k++)
            {
                DS[l].x[k] = Core[l % NumCluster].x[k] + (double) rand() / ((double) RAND_MAX +
1.0) * CLUSTER_DISTANCE;
            }
            DataClass[l] = l % NumCluster;
            l++;
        }
        if(Core != NULL)
        {
            free(Core);
            Core = NULL;
        }
```

```
    return 0;
}
```

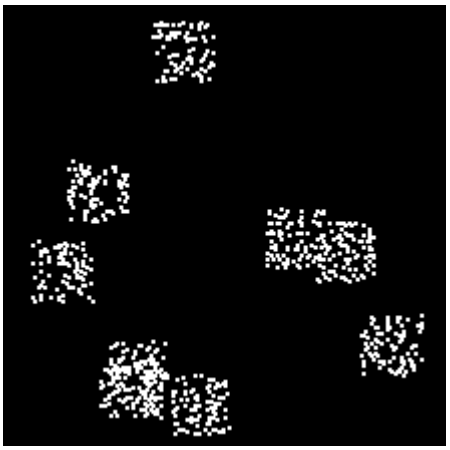
Online Resource 4. Other figures and tables of experimental results

4.2 Experimental results of some artificial data sets

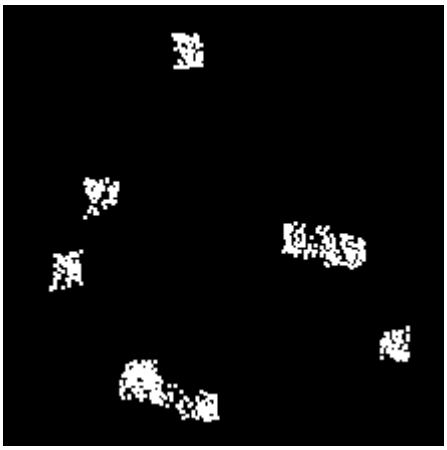
4.2.1 Compare the dynamic clustering processes of SynC algorithm, ESynC algorithm and SSynC algorithm



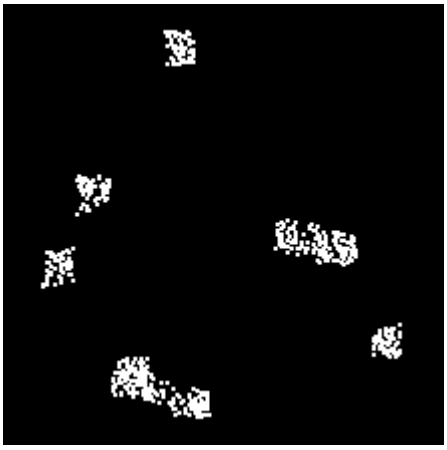
(a) $t = 0$ (The original locations of 800 data points created from DataType2)



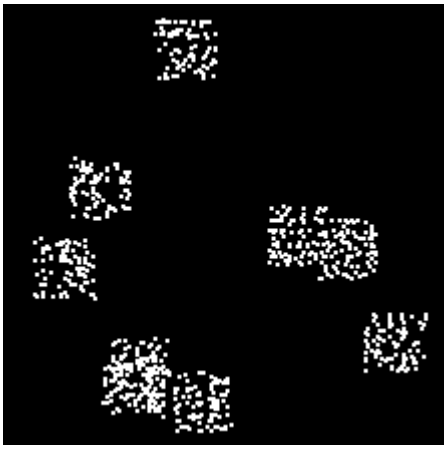
(b-1) SynC algorithm, $t = 1$



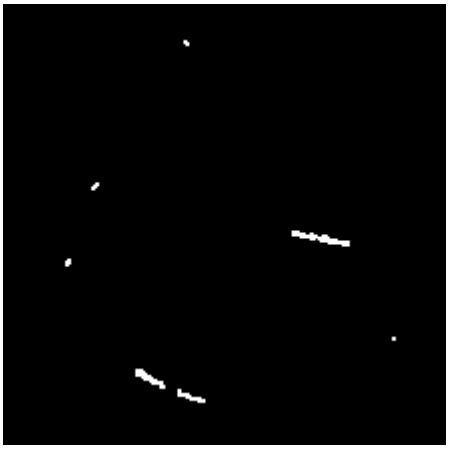
(b-2) ESynC algorithm, $t = 1$



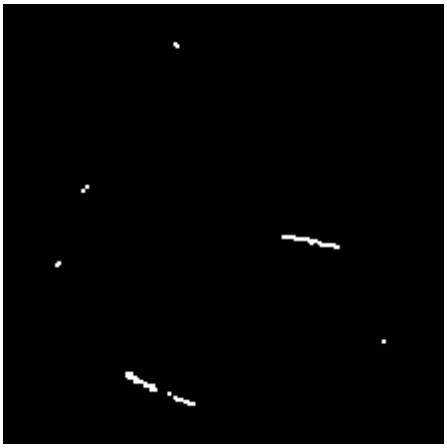
(b-3) SSynC algorithm, $t = 1$



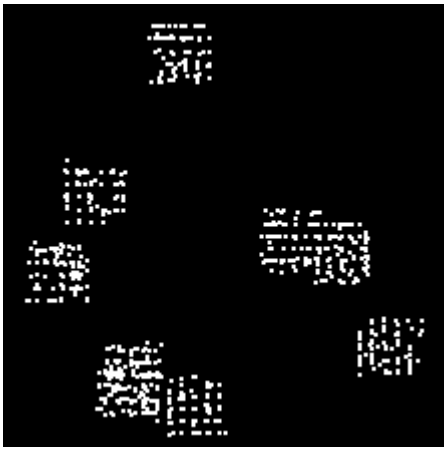
(c-1) SynC algorithm, $t = 2$



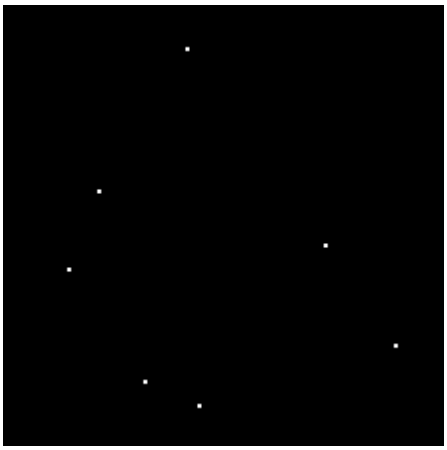
(c-2) ESynC algorithm, $t = 2$



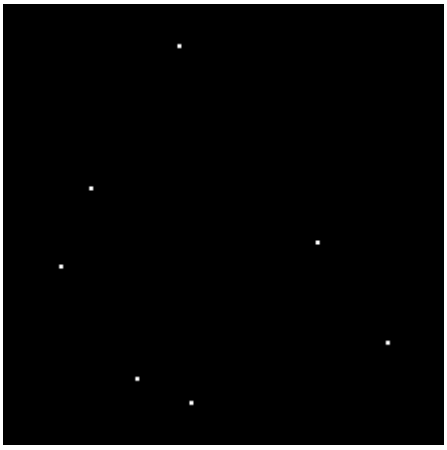
(c-3) SSynC algorithm, $t = 2$



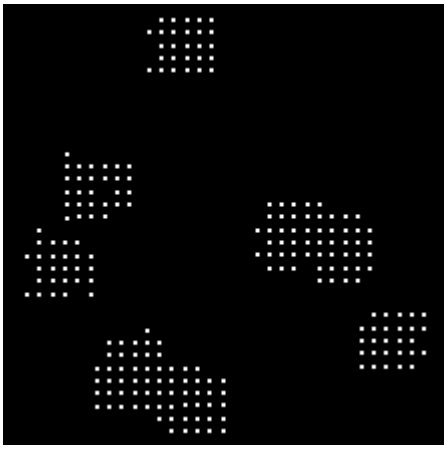
(d-1) SynC algorithm, $t = 5$



(d-2) ESynC algorithm, $t = 5$

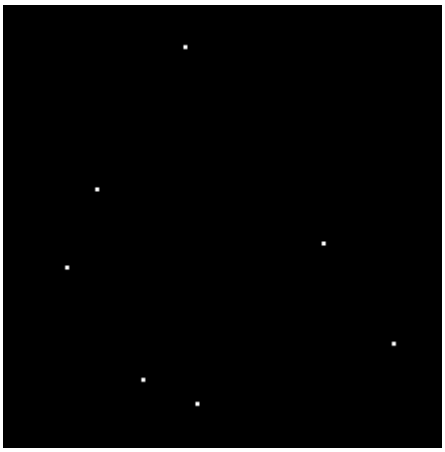


(d-2) SSynC algorithm, $t = 5$



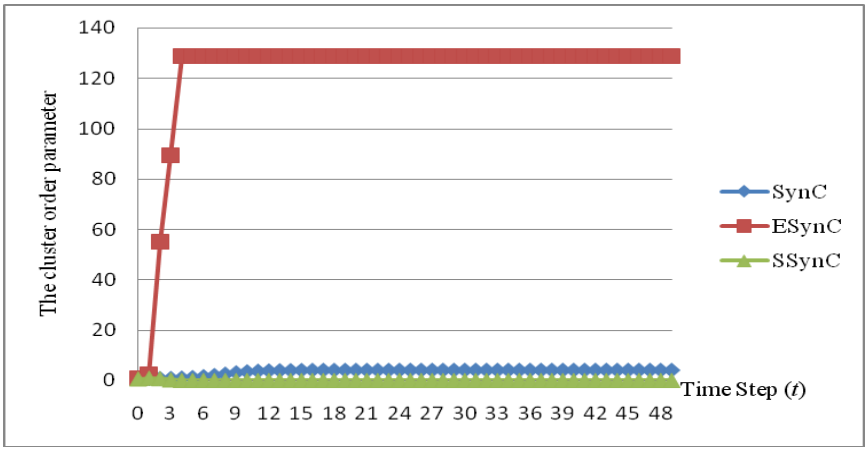
(e-1) SynC algorithm, $t = 45$

(e-2) ESynC algorithm, $t = 45$

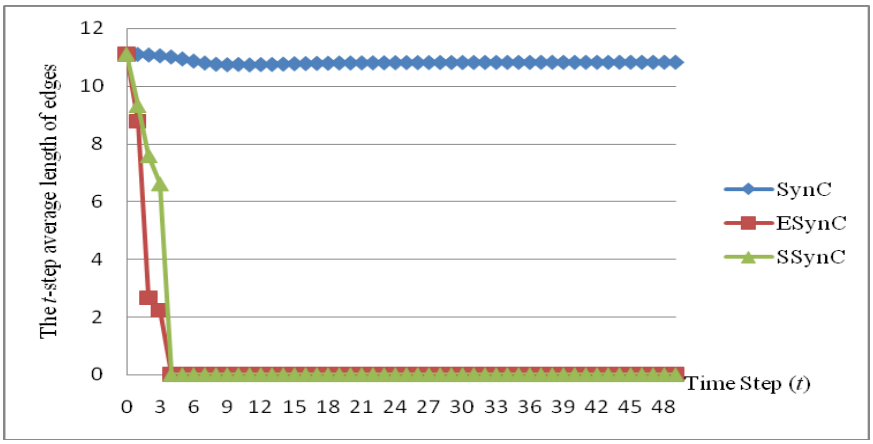


(e-3) SSynC algorithm, $t = 45$

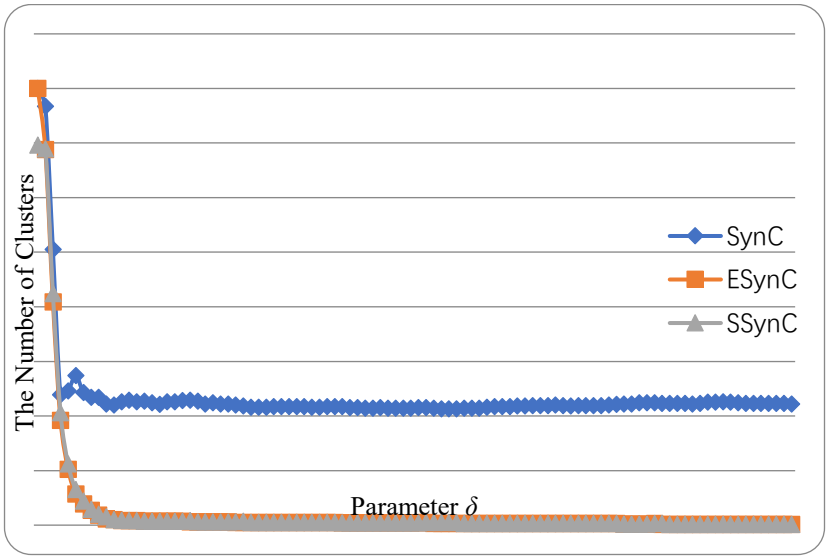
Sfig. 1. Compare the dynamical synchronization clustering processes with time evolution among SynC algorithm, ESynC algorithm and SSynC algorithm. From (a) to (e) of Sfig. 1, the data set has 800 points created from DataType2, parameter δ is set as 18 in the three algorithms, and parameter ε is set as 1 in SSynC algorithm.



(a) The cluster order parameter r_c with t-step evolution (t: 0 - 49)



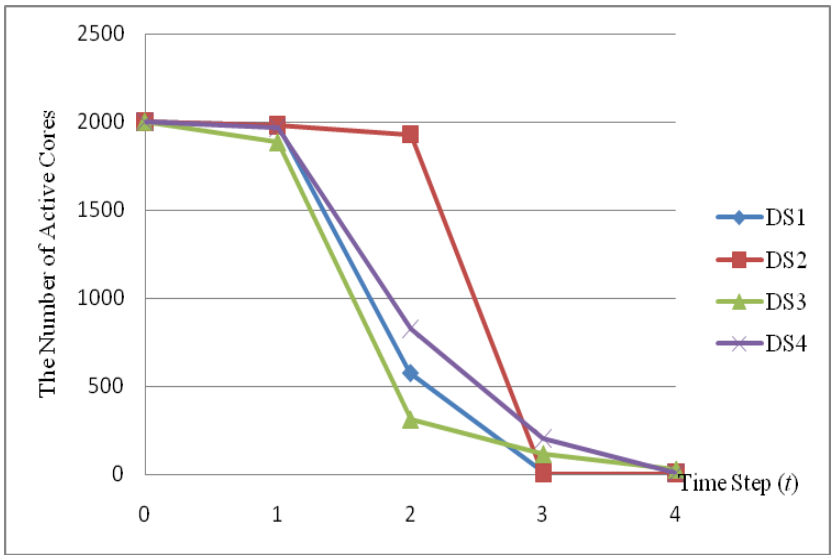
(b) The t-step average length of edges AveLen(t) (t: 0 - 49)



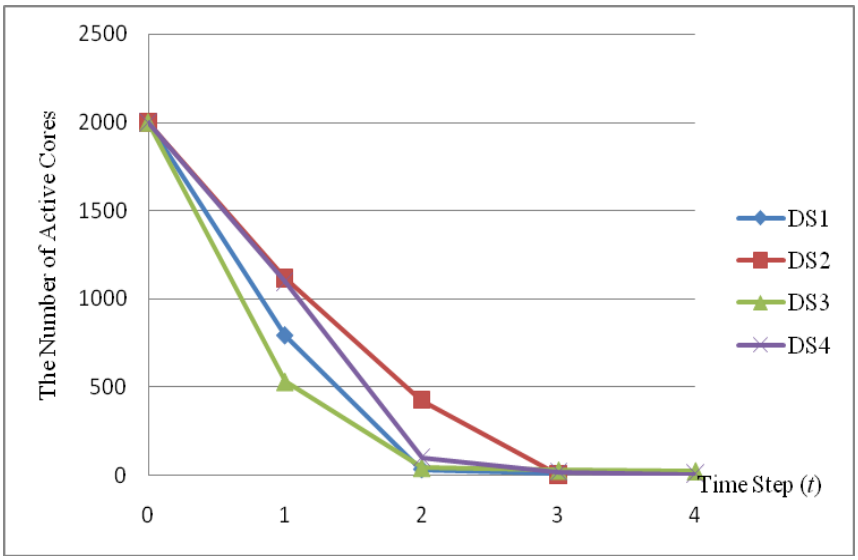
(c) The relation between the final number of clusters and parameter δ (δ: 0 - 99).

Sfig. 2. Compare SynC algorithm, ESynC algorithm and SSynC algorithm. In Sfig. 2, the data set has 800 points created from DataType2, and parameter ε is set as 1 in SSynC algorithm. In Sfig.2 (a) and (b), parameter δ is set as 18 in the three algorithms.

4.2.3 Setting parameters in SSynC algorithm

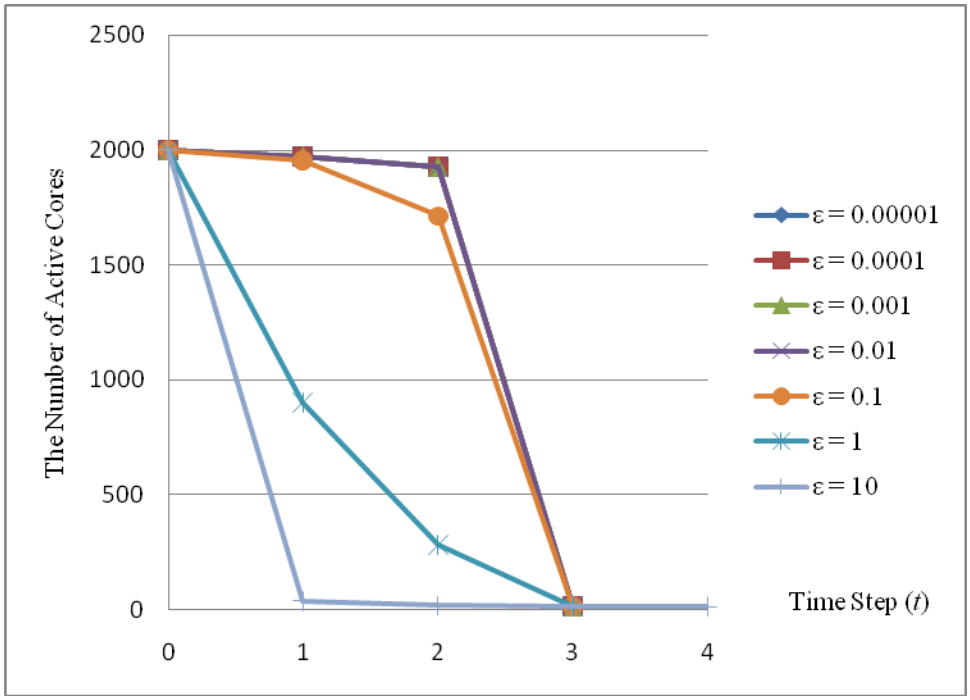


(a) Parameter $\varepsilon = 0.00001$

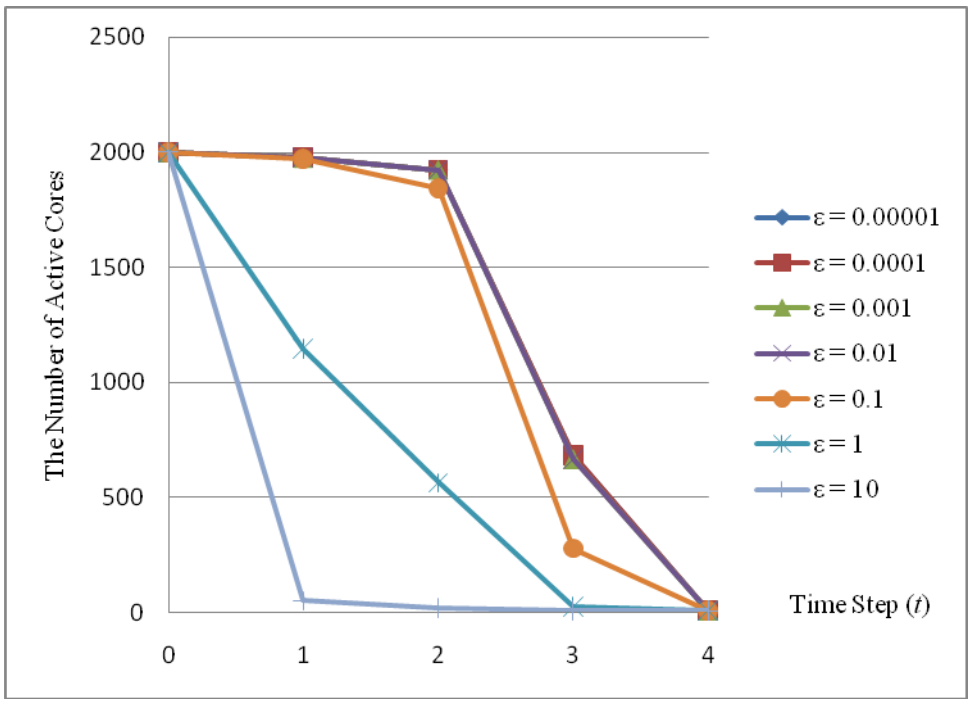


(b) Parameter $\varepsilon = 1$

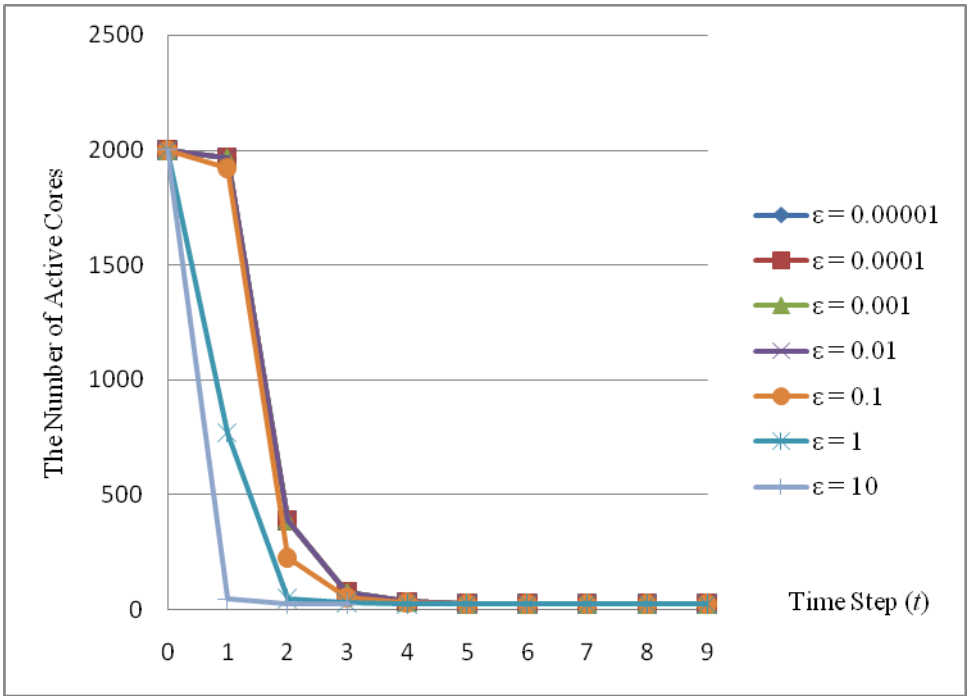
Sfig. 3. The number of active cores with time evolution of four datasets in SSynC algorithm. In Sfig. 3, parameter δ is set as 22, parameter ε is set as 0.00001 and 1, and the numbers of points in the four data sets are all set as 2000.


2500
2000
1500
1000
500
0
The Number of Active Cores
0
1
2
3
4
Time Step (t)
ε = 0.00001
ε = 0.0001
ε = 0.001
ε = 0.01
ε = 0.1
ε = 1
ε = 10

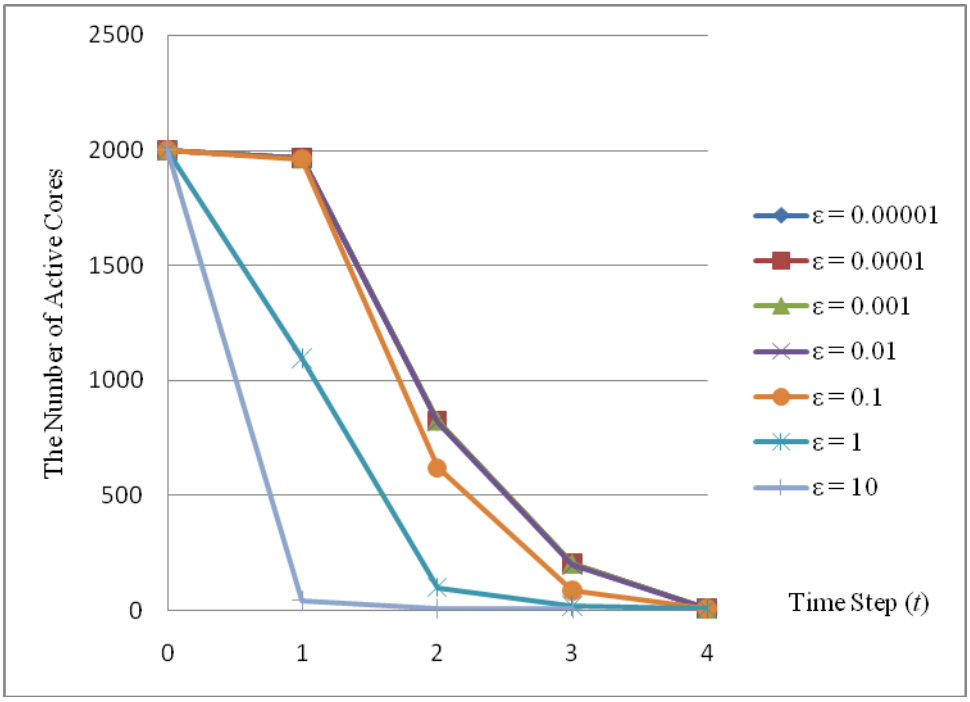
(a) DS1


2500
2000
1500
1000
500
0
The Number of Active Cores
0
1
2
3
4
Time Step (t)
ε = 0.00001
ε = 0.0001
ε = 0.001
ε = 0.01
ε = 0.1
ε = 1
ε = 10

(b) DS2

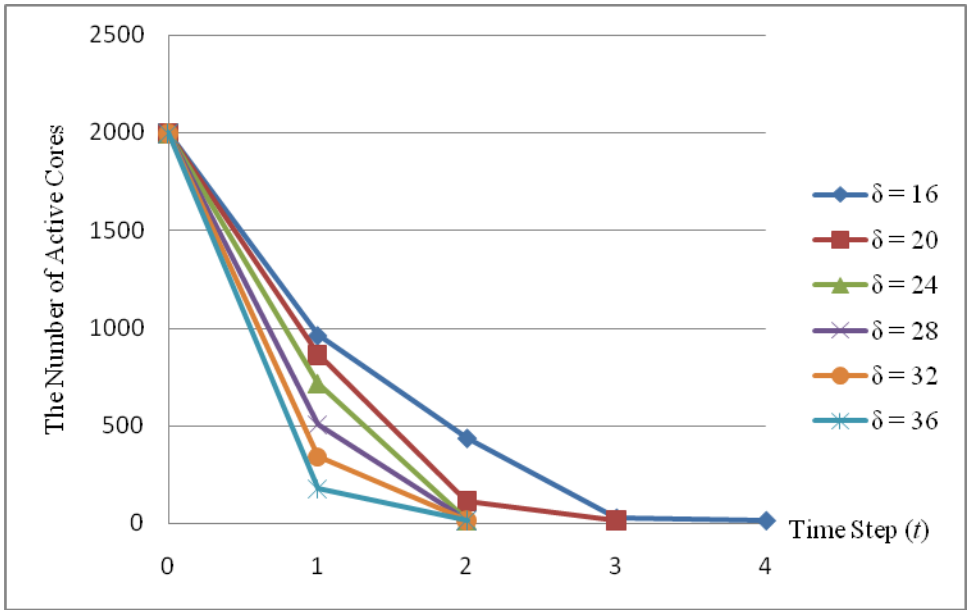


(c) DS3

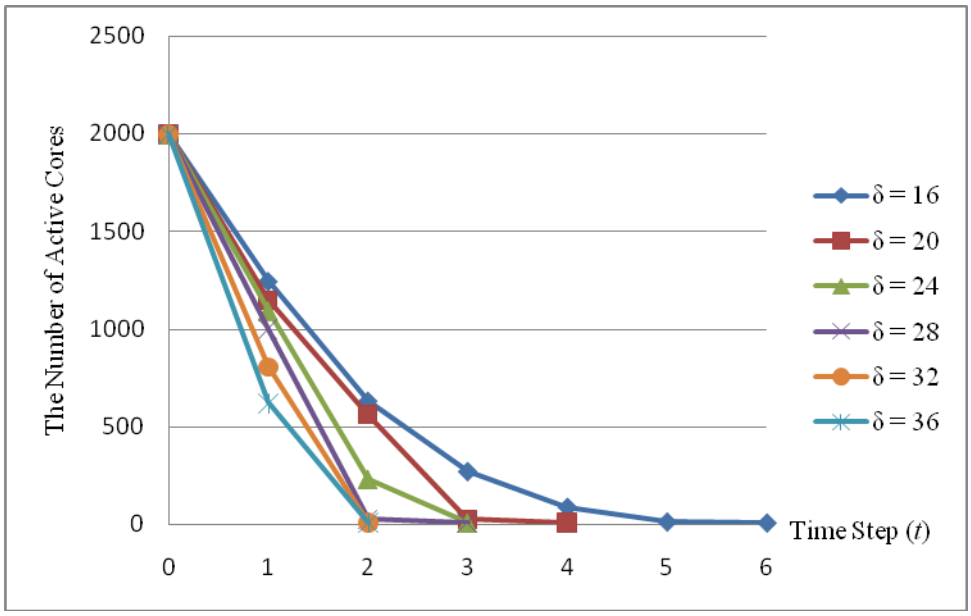


(d) DS4

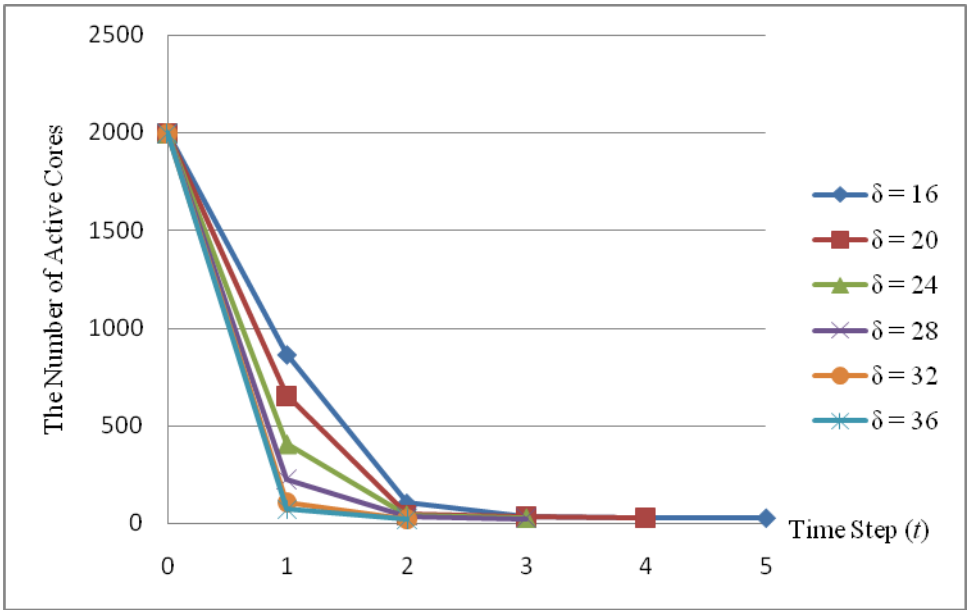
Sfig. 4. The number of active cores with time evolution for several different values of parameter ε in SSynC algorithm. In Sfig. 4, parameter $\delta = 18$ in DS1 and DS3, $\delta = 20$ in DS2, $\delta = 22$ in DS4; parameter ε is set as seven different value (0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10) respectively; and the number of points in the four data sets is all set as 2000.


2500
2000
1500
1000
500
0
The Number of Active Cores
0
1
2
3
4
Time Step (t)
δ = 16
δ = 20
δ = 24
δ = 28
δ = 32
δ = 36

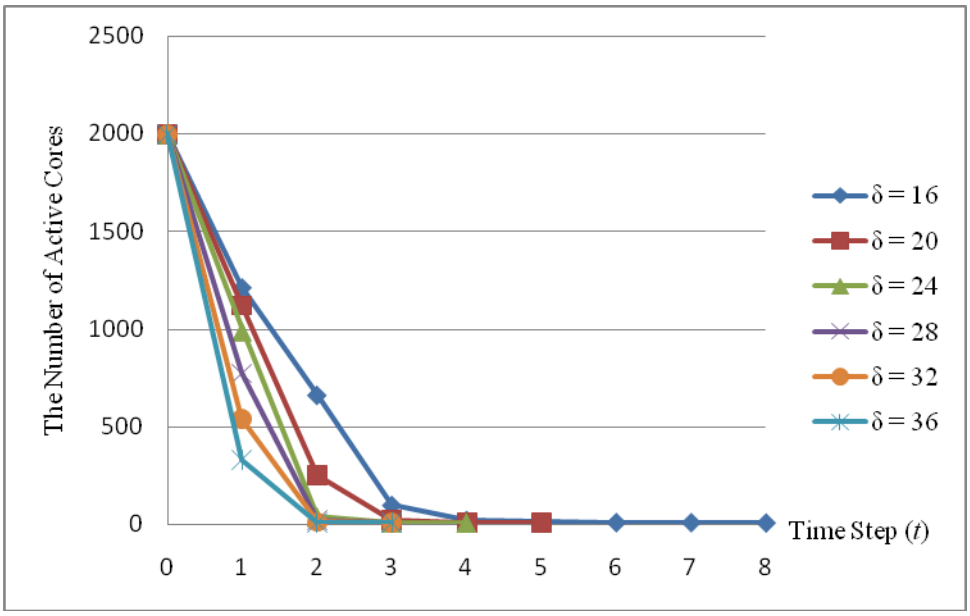
(a) DS1


2500
2000
1500
1000
500
0
The Number of Active Cores
0
1
2
3
4
5
6
Time Step (t)
δ = 16
δ = 20
δ = 24
δ = 28
δ = 32
δ = 36

(b) DS2



(c) DS3



(d) DS4

Sfig. 5. The number of active cores with time evolution for several different values of parameter δ in SSynC algorithm. In Sfig.5, parameter δ is set as six different values (16, 20, 24, 28, 32, 36) respectively, parameter ε is set as 1, and the number of points in the four data sets is all set as 2000.

4.2.4 Compare the clustering results among SynC algorithm, ESynC algorithm, SSynC algorithm and some classical clustering algorithms

Stable 4. Comparison of three different synchronization algorithms (SynC, ESynC and SSynC) by using four artificial data sets (DS1 – DS4). In Stable 4, parameter $\delta = 18$, the number of data points $n = 10000$, parameter ε (a very small real number, if the distance of two points is less than ε, then they are regarded as in the same cluster) = 0.00001 in SSynC algorithm.

Comparison of Algorithms		**DS1**	**DS2**	**DS3**	**DS4**
Spend time (second)	SynC	448	553	538	525
	ESynC	56	70	107	81
	SSynC	**52**	**69**	**34**	**52**
Iterative times	SynC	41	50	50	50
	ESynC, SSynC	**4**	**5**	**8**	**6**
The number of steady locations	SynC	254	379	260	431
	ESynC, SSynC	**14**	**5**	**25**	**8**

Note: The bold in Stable 4 marks the better results of SSynC algorithm or ESynC algorithm.

Stable 5. Compare the clustering quality of several clustering algorithms (SynC, ESynC, SSynC, and some classical clustering algorithms) using six kinds of artificial data sets (DS2, DS4, DS5, DS6, DS7, and DS8). In Stable 5, parameter $\delta = 18$ in DS2, DS4, DS5, and DS6; parameter $\delta = 30$ in DS7 and DS8; parameter $\varepsilon = 0.00001$ in SSynC algorithm.

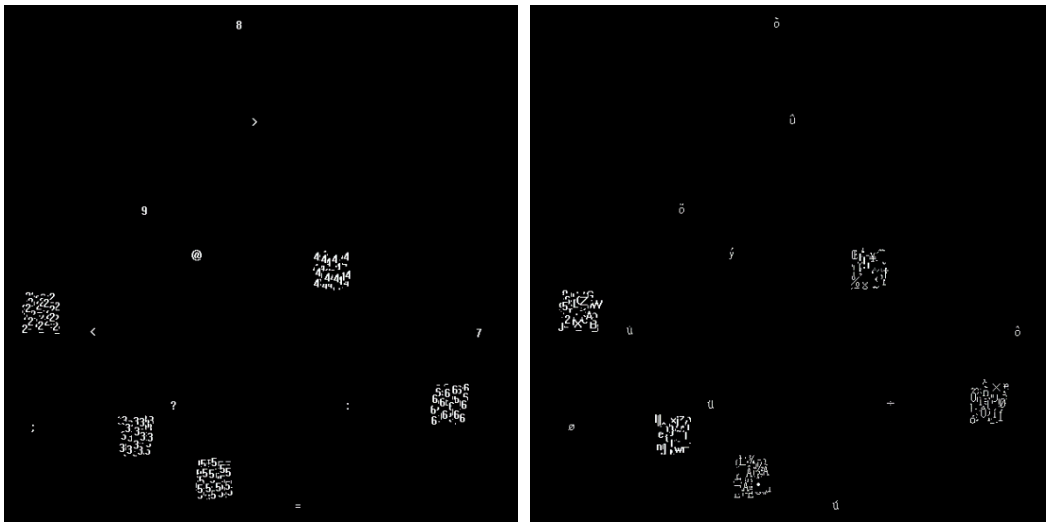
(a)

Comparison of Clustering Algorithms		DS2 (n = 400)	DS4 (n = 400)	DS2 (n = 800)	DS4 (n = 800)
NMI	SSynC, ESynC	**1.0000**	0.9694	**1.0000**	0.9643
	SynC	0.5505	0.6324	0.5362	0.6099
	K-Means	0.8670	0.9185	0.8659	**0.9682**
	FCM	**1.0000**	0.9633	**1.0000**	0.9615
	AP	0.7966	**0.9697**	0.7355	0.8375
	DBSCAN	**1.0000**	0.9643	**1.0000**	0.9643
	Mean Shift	0.7978	0.9028	0.7799	0.9103
AMI	SSynC, ESynC	**1.0000**	0.9682	**1.0000**	0.9286
	SynC	0.1237	0.1275	0.1653	0.1785
	K-Means	0.8255	0.8980	0.8266	**0.9676**
	FCM	**1.0000**	0.9616	**1.0000**	0.9603
	AP	0.6252	**0.9684**	0.5333	0.7157
	DBSCAN	**1.0000**	0.9274	**1.0000**	0.9286
	Mean Shift	0.6251	0.8268	0.6022	0.8758
The Number of Clusters	SSynC, ESynC	**5**	**9**	**5**	**8**
	SynC	227	255	314	357
	K-Means	5 (predefined)	9 (predefined)	5 (predefined)	9 (predefined)
	FCM	5 (predefined)	9 (predefined)	5 (predefined)	9 (predefined)
	AP	13	**9**	20	19
	DBSCAN	**5**	**8**	**5**	**8**
	Mean Shift	15	15	17	14

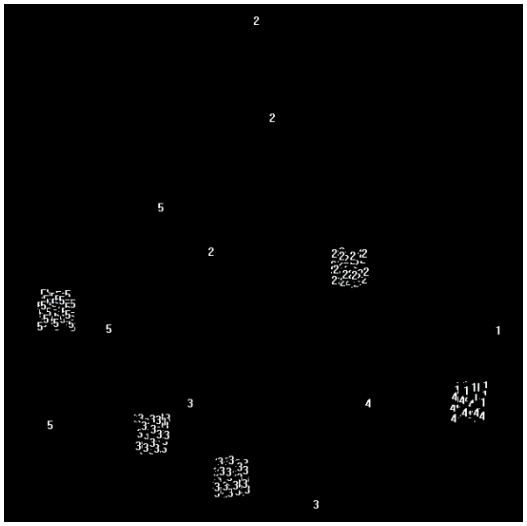
(b)

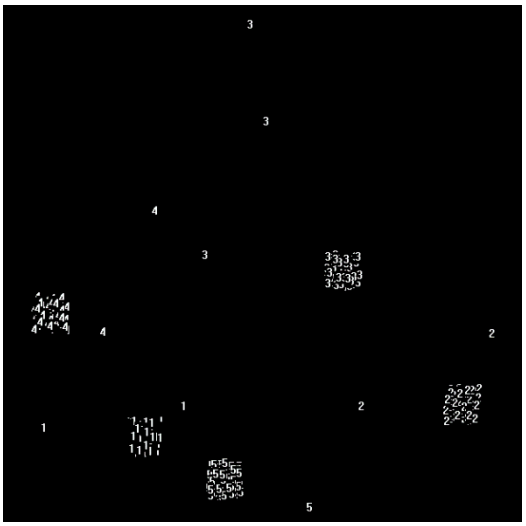
Comparison of Clustering Algorithms		DS5 (n = 10000)	DS6 (n = 10000)	DS7 (n = 10000)	DS8 (n = 10000)
NMI	SSynC, ESynC	0.9765	**1.0000**	**1.0000**	**1.0000**
	SynC	0.6231	0.5411	0.5205	0.5194
	K-Means	0.8872	NaN (Matlab)	0.9194	0.8437
	FCM	**0.9788**	0.5228	0.5226	0.5282
	DBSCAN	0.9765	**1.0000**	**1.0000**	**1.0000**
	Mean Shift	0.9708	**1.0000**	**1.0000**	**1.0000**
AMI	SSynC, ESynC	0.9534	**1.0000**	**1.0000**	**1.0000**
	SynC	0.3539	0.0973	0.0051	1.5118e-04
	K-Means	0.8426	NaN (Matlab)	0.8892	0.7783
	FCM	**0.9781**	0.5228	0.5226	0.2788
	DBSCAN	0.9534	**1.0000**	**1.0000**	**1.0000**
	Mean Shift	0.9534	**1.0000**	**1.0000**	**1.0000**
The Number of Clusters	SSynC, ESynC	**11**	**12**	**12**	**12**
	SynC	578	5577	9729	9992
	K-Means	12 (predefined)	1 (+11 null clusters)	12 (predefined)	12 (predefined)
	FCM	12 (predefined)	2 (+10 null clusters)	3 (+9 null clusters)	2 (+10 null clusters)
	DBSCAN	**11**	**12**	**12**	**12**
	Mean Shift	**12**	**12**	**12**	**12**

Note: NMI and AMI are two clustering quality measures presented in Vinh et al.(2010). In Stable 5, the largest values of NMI and AMI and acceptable number of clusters in every data set are shown in bold.

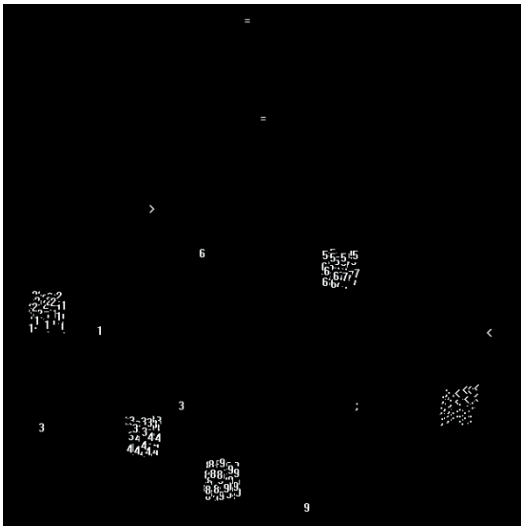


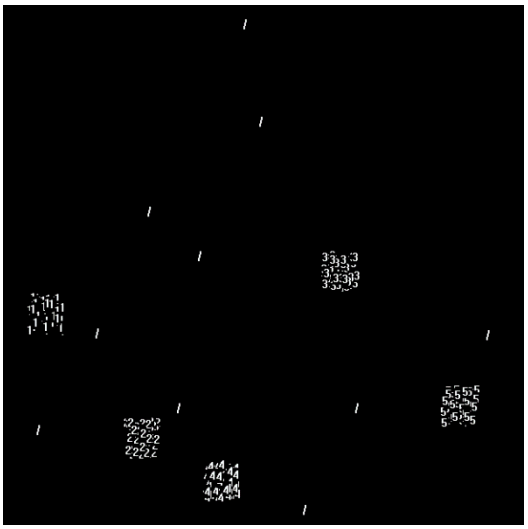
(a) Clusters identified by ESynC (15 clusters or isolates) (b) Clusters identified by SynC (204 clusters or isolates)



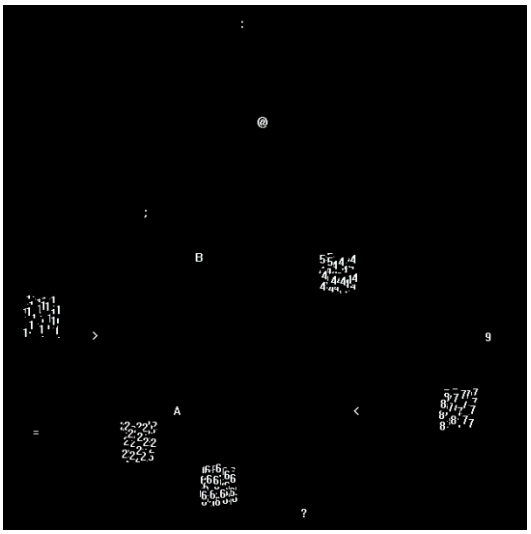


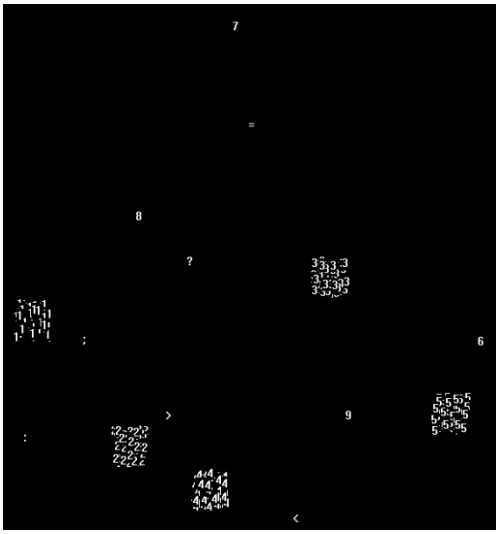
(c) Clusters identified by KMeans (predefined 5 clusters) (d) Clusters identified by FCM (predefined 5 clusters)





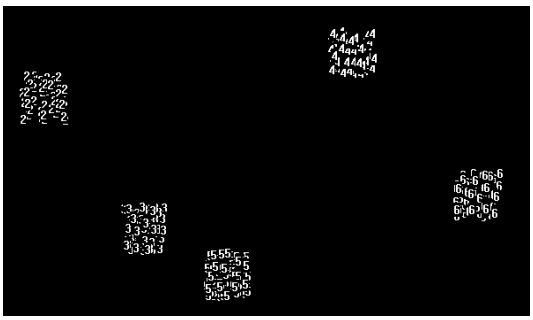
(e) Clusters identified by AP (14 clusters) (f) Clusters identified by DBSCAN (5 clusters)



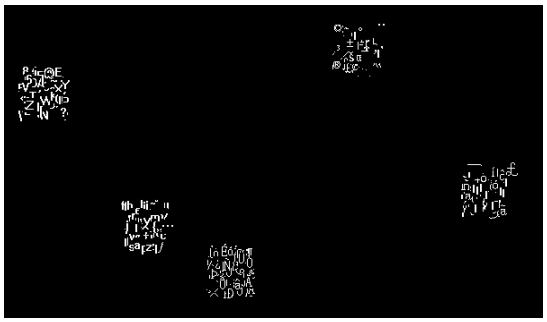


(g) Clusters identified by Mean Shift (18 clusters) (f) Clusters identified by SSynC (15 clusters or isolates)

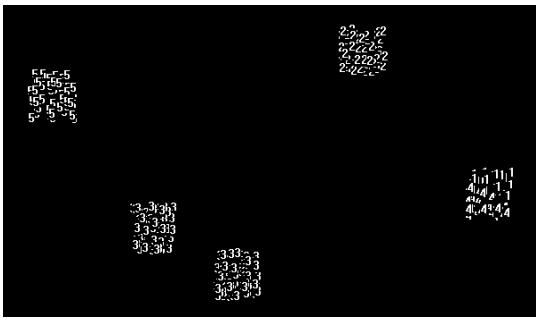
Sfig. 6. Compare the clustering results of several algorithms (DS1, n = 400)



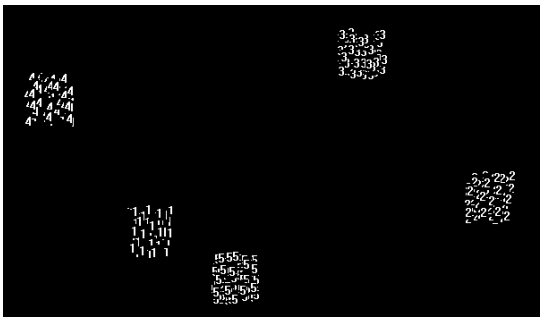
(a) Clusters identified by ESynC (5 clusters)



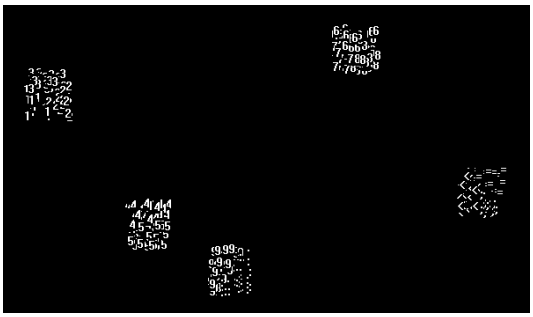
(b) Clusters identified by SynC (227 clusters)



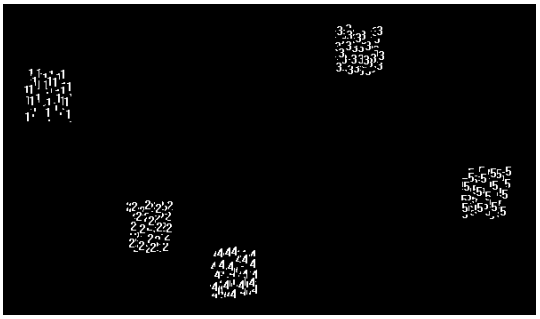
(c) Clusters identified by KMeans (predefined 5 clusters)



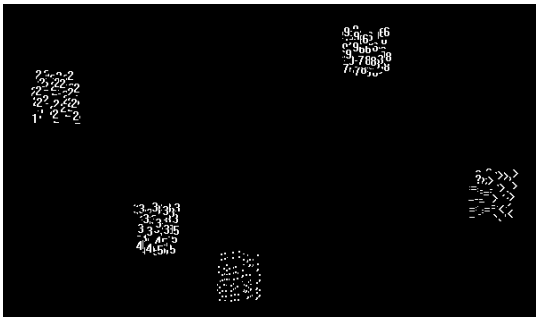
(d) Clusters identified by FCM (predefined 5 clusters)



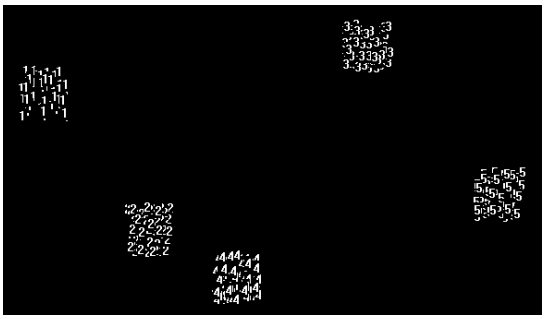
(e) Clusters identified by AP (13 clusters)



(f) Clusters identified by DBSCAN (5 clusters)

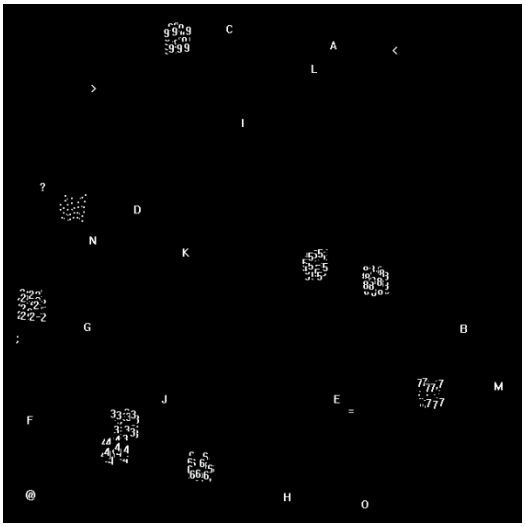


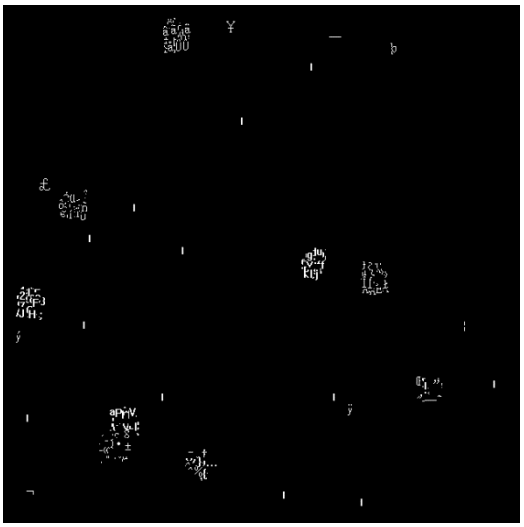
(g) Clusters identified by Mean Shift (15 clusters)



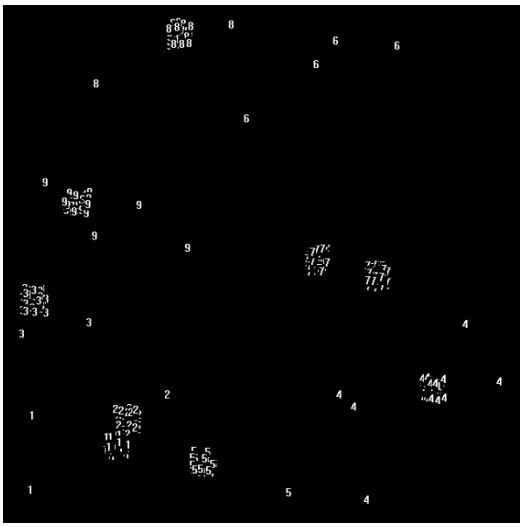
(f) Clusters identified by SSynC (5 clusters)

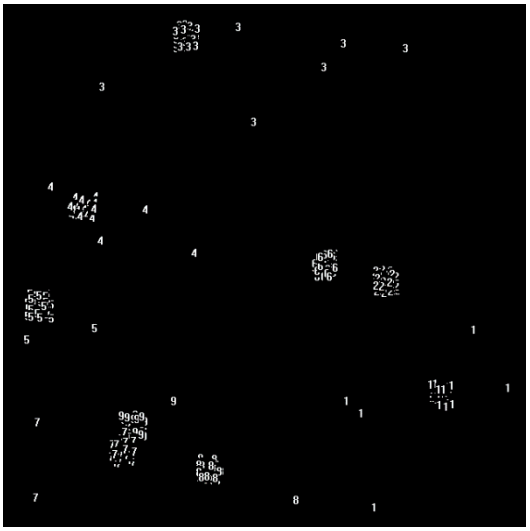
Sfig. 7. Compare the clustering results of several algorithms (DS2, n = 400)



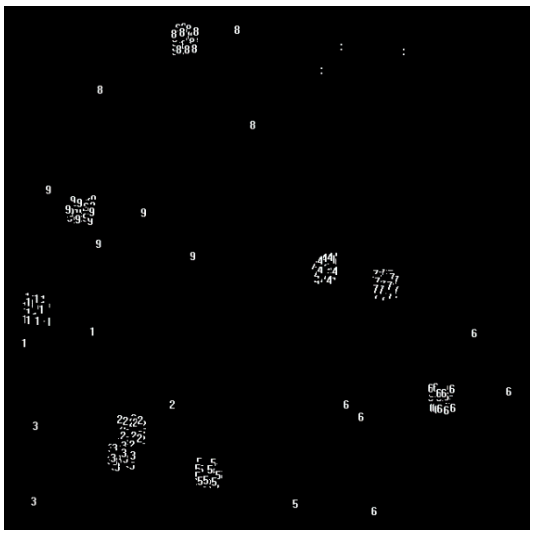


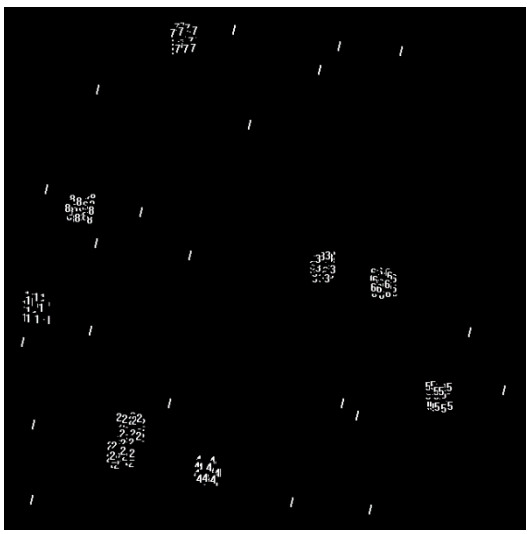
(a) Clusters identified by ESynC (30 clusters or isolates) (b) Clusters identified by SynC (224 clusters or isolates)



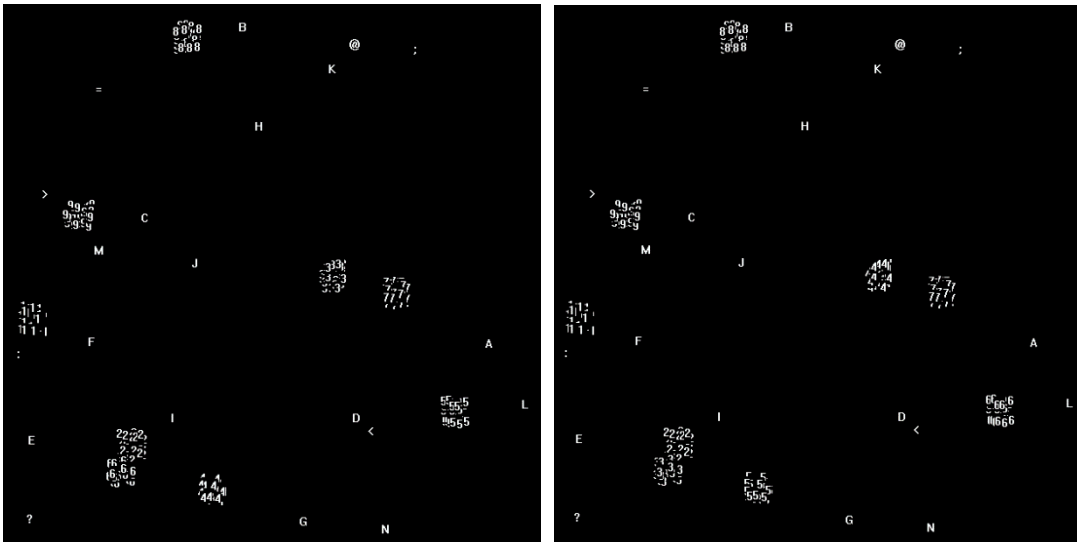


(c) Clusters identified by KMeans (predefined 9 clusters) (d) Clusters identified by FCM (predefined 9 clusters)



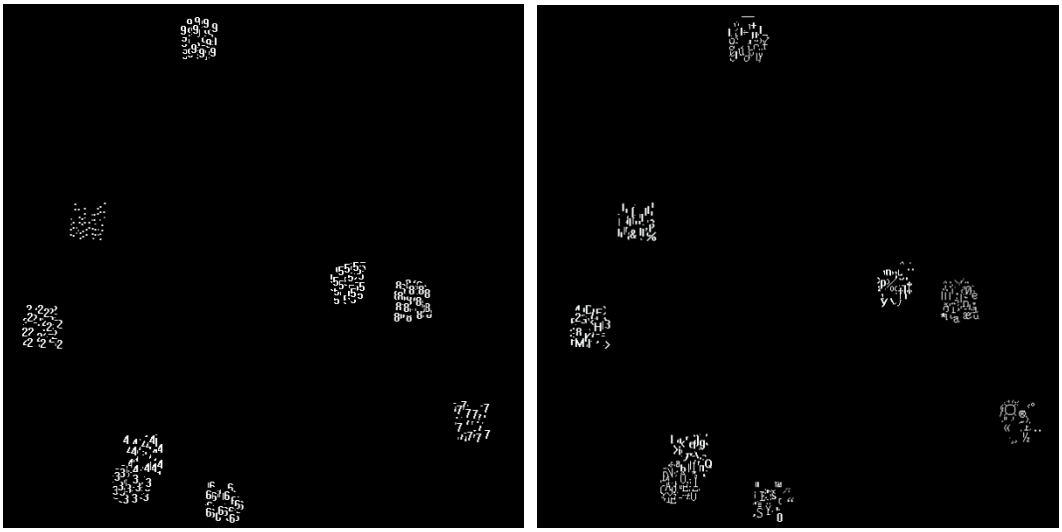


(e) Clusters identified by AP (10 clusters) (f) Clusters identified by DBSCAN (8 clusters)

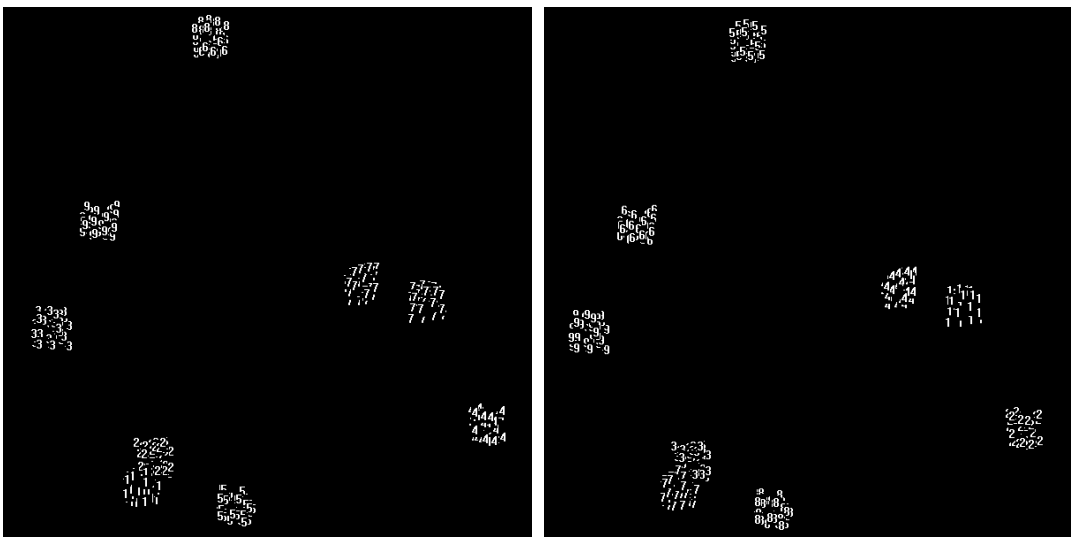


(g) Clusters identified by Mean Shift (30 clusters) (f) Clusters identified by SSynC (30 clusters or isolates)

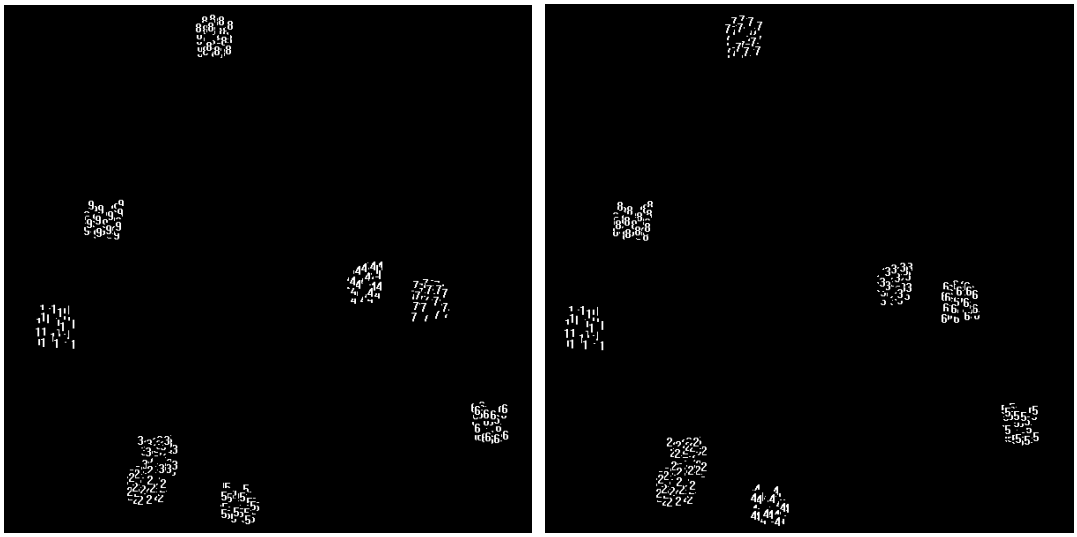
Sfig. 8. Compare the clustering results of several algorithms (DS3, n = 400)



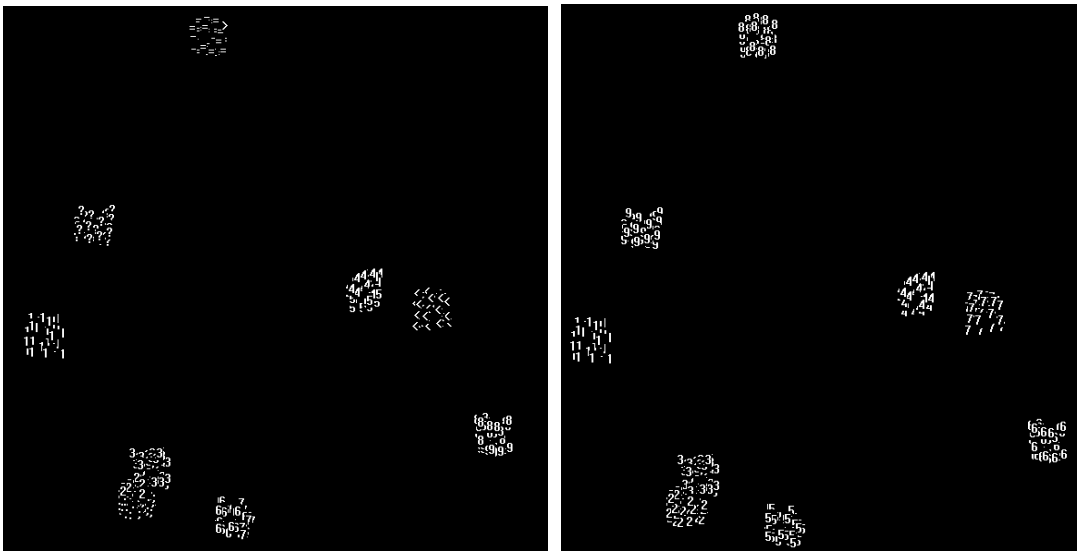
(a) Clusters identified by SSynC and ESynC (9 clusters) (b) Clusters identified by SynC (255 clusters)



(c) Clusters identified by KMeans (predefined 9 clusters) (d) Clusters identified by FCM (predefined 9 clusters)

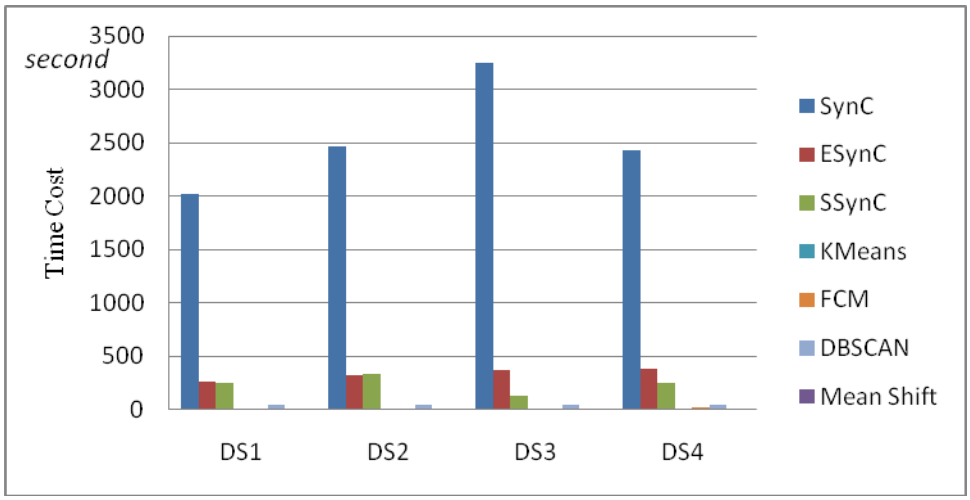


(e) Clusters identified by AP (9 clusters) (f) Clusters identified by DBSCAN (8 clusters)



(g) Clusters identified by Mean Shift (15 clusters) (f) Clusters identified by SSynC (9 clusters)

Sfig. 9. Compare the clustering results of several algorithms (DS4, n = 400)



Sfig. 10. Comparison of several algorithms in time cost by using four artificial data sets (DS1 - DS4, n

= 20000).

Stable 6. Compare the valid interval of parameter δ among SynC, ESynC, SSynC, DBSCAN, and Mean Shift using some artificial data sets with different dimensions. In Stable 6, n = 10000, parameter ε = 0.00001 in SSynC algorithm.

(a)

The valid interval of parameter δ	DS5	DS6	DS7	DS8
SynC	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$
SSynC, ESynC	$\delta \in$ [9, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
DBSCAN	$\delta \in$ [2, 45]	$\delta \in$ [7, 147]	$\delta \in$ [12, 199]	$\delta \in$[17, 281]
Mean Shift	$\delta \in$[15, 60]	$\delta \in$[17, 176]	$\delta \in$[20, 285]	$\delta \in$[22, 396]
$[e_k , e_{k+1}]$ In MST	[2.16, 45.42]	[9.82, 147.48]	[15.29, 199.78]	[21.04, 281.19]

(b)

The valid interval of parameter δ	DS9	DS10	DS11	DS12
SynC	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$
SSynC, ESynC	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
DBSCAN	$\delta \in$ [2, 68]	$\delta \in$ [6, 193]	$\delta \in$ [11, 232]	$\delta \in$[15, 279]
Mean Shift	$\delta \in$[14, 89]	$\delta \in$[15, 219]	$\delta \in$[19, 261]	$\delta \in$[21, 312]
$[e_k , e_{k+1}]$ In MST	[1.36, 68.69]	[6.89, 193.04]	[6.89, 193.04]	[18.47, 279.44]

(c)

The valid interval of parameter δ	DS13	DS14	DS15	DS16
SynC	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$	$\delta \in \emptyset$
SSynC, ESynC	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1850]	$\delta \in$ [123, 2917]
DBSCAN	$\delta \in$ [34, 841]	$\delta \in$ [57, 1257]	$\delta \in$ [90, 1841]	$\delta \in$[135, 2908]
Mean Shift	$\delta \in$[40, 872]	$\delta \in$[65, 1283]	$\delta \in$[92, 1864]	$\delta \in$[136, 2935]
$[e_k , e_{k+1}]$ In MST	[39.69, 841.37]	[64.05, 1257.35]	[97.34, 1841.97]	[142.44, 2908.82]

Stable 7. Compare the valid interval of parameter δ in SSynC algorithm for several different value of parameter ε using some artificial data sets with different dimensions. In Stable 7, n = 10000, parameter ε is set as several different value respectively in SSynC algorithm.

(a) DS5 – DS8

SSynC	DS5	DS6	DS7	DS8
ε = 0.00001	$\delta \in$ [9, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
ε = 0.0001	$\delta \in$ [9, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
ε = 0.001	$\delta \in$ [9, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
ε = 0.01	$\delta \in$ [9, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
ε = 0.1	$\delta \in$ [13, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 214]	$\delta \in$[22, 298]
ε = 1	$\delta \in$ [12, 58]	$\delta \in$ [11, 164]	$\delta \in$ [16, 215]	$\delta \in$[22, 298]
ε = 10	$\delta \in$ [14, 22]	$\delta \in$ [16, 161]	$\delta \in$ [17, 215]	$\delta \in$[22, 298]

(b) DS9 – DS12

SSynC	DS9	DS10	DS11	DS12
ε = 0.00001	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 0.0001	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 0.001	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 0.01	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 0.1	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 1	$\delta \in$ [9, 83]	$\delta \in$ [10, 208]	$\delta \in$ [13, 248]	$\delta \in$[19, 297]
ε = 10	$\delta \in$ [14, 83]	$\delta \in$ [16, 208]	$\delta \in$ [16, 249]	$\delta \in$[19, 297]

(c) DS13 – DS16

SSynC	DS13	DS14	DS15	DS16
ε = 0.01	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1850]	$\delta \in$ [123, 2917]
ε = 0.1	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1850]	$\delta \in$ [123, 2917]
ε = 1	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1851]	$\delta \in$ [123, 2917]
ε = 10	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1851]	$\delta \in$ [123, 2917]
ε = 20	$\delta \in$ [40, 854]	$\delta \in$ [63, 1271]	$\delta \in$ [87, 1851]	$\delta \in$ [123, 2918]
ε = 30	$\delta \in$ [41, 854]	$\delta \in$ [64, 1271]	$\delta \in$ [87, 1851]	$\delta \in$ [125, 2919]
ε = 40	$\delta \in$ [42, 854]	$\delta \in$ [65, 1271]	$\delta \in$ [89, 1851]	$\delta \in$ [125, 2917]

Note: SSynC algorithm gets 12 clusters when parameter δ in its valid interval**.** In the DS5 (n = 10000) data set, there are two clusters that are almost connected to one cluster, so parameter ε affects the final number of clusters very much. For other data sets, parameter ε affects the final number of clusters little.

4.3 Experimental results of several UCI data sets

Stable 8. Compare three synchronization algorithms (SynC, ESynC and SSynC) by using several UCI data sets. In Stable 8, parameter $\varepsilon = 1$ in SSynC algorithm.

(a) The setting of parameter δ in three synchronization algorithms for several UCI data sets

UCI Data Sets	parameter δ in SynC, ESynC and SSynC
Iris	120
Wine	305
Wdbc	345
Glass	148
Ionosphere	615
Letter-recognition	210
Segmentation	205
Cloud	380

(b) Comparison results of the first four UCI data sets

Comparison of Algorithms		Iris	Wine	Wdbc	Glass
Spend time (second)	SynC	0	0	15	0
	ESynC	0	0	2	0
	SSynC	0	0	**1**	0
Iterative times	SynC	50	50	50	50
	SSynC, ESynC	**9**	**6**	**7**	**6**
The number of steady locations	SynC	147	178	569	213
	SSynC, ESynC	**5**	**19**	**35**	**35**
The cluster order parameter r_c	SynC	0.05333	0	0	0.009346
	ESynC	**54.1067**	**47.8876**	**305.3497**	**55.1402**
	SSynC	**0**	**0**	**0**	**0**
AveLen(T)	SynC	83.9640	258.3664	276.6775	97.9706
	SSynC, ESynC	**0**	**0**	**0**	**0**

(c) Comparison results of the next four UCI data sets

Comparison of Algorithms		Ionosphere	Letter-recognition	Segmentation	Cloud
Spend time (second)	SynC	5	4186	1	79
	ESynC	1	2270	0	10
	SSynC	**1**	**394**	**0**	**4**
Iterative times	SynC	50	50	50	50
	SSynC, ESynC	**9**	**23**	**7**	**6**
The number of steady locations	SynC	350	18668	210	2043
	SSynC, ESynC	**85**	**34**	**38**	**2**
The cluster order parameter r_c	SynC	0.005698	0.2596	0.000036	0.004965
	ESynC	**126.49**	**9107.0009**	**19.5905**	**1023**
	SSynC	**0**	**0**	**0**	**0**
AveLen(T)	SynC	401.6912	171.9401	142.6595	215.9900
	SSynC, ESynC	**0**	**0**	**0**	**0**

Note: The bold in Stable 8 marks the better results of SSynC algorithm or ESynC algorithm.

Stable 9. Compare the clustering quality of several clustering algorithms (SynC, ESynC, SSynC and some classical clustering algorithms) by using several UCI data sets. In Stable 9, parameter $\varepsilon = 1$ in SSynC algorithm.

(a) The setting of parameter δ in several clustering algorithms for several UCI data sets

UCI Data Sets	parameter δ in SynC, ESynC and SSynC	parameter δ in DBSCAN	parameter δ in Mean Shift
Iris	120	75	150
Wine	305	242.725	305
Wdbc	345	215	345
Glass	148	80	120
Ionosphere	615	350	710
Letter-recognition	210	160	220
Segmentation	205	176	270
Cloud	380	350	350

(b) Comparison results of the first four UCI data sets

Comparison of Clustering Algorithms		Iris	Wine	Wdbc	Glass
NMI	SSynC, ESynC	0.7265	0.7615	0.4655	0.4540
	SynC	0.4697	0.4578	0.3226	**0.5306**
	K-Means	0.7145	**0.8782**	**0.6232**	0.3588
	FCM	**0.7919**	0.4823	0.5947	0.4108
	AP	0.6061	0.5382	0.3594	0.4257
	DBSCAN	0.6465	0.3534	0.2904	0.2574
	Mean Shift	0.7265	0.7612	0.2797	0.4662
AMI	SSynC, ESynC	0.7143	0.6057	0.3513	0.2872
	SynC	0.0050.	3.2528e-16	6.8369e-16	0.0012
	K-Means	0.7107	**0.8735**	**0.6110**	**0.3265**
	FCM	**0.7888**	0.3820	0.5887	0.2525
	AP	0.3982	0.2977	0.1453	0.2423
	DBSCAN	0.5712	0.3423	0.2496	0.2065
	Mean Shift	0.7143	0.5819	0.2086	0.2414
The Number of Clusters	SSynC, ESynC	3 (+ 2 isolates)	3 (+ 16 isolates)	2 (+ 33 isolates)	6 (+29 isolates)
	SynC	2 (+ 145 isolates)	0 (+178 isolates)	0 (+ 569 isolates)	1 (+ 212 isolates)
	K-Means	3 (predefined)	3 (predefined)	2 (predefined)	6 (predefined)
	FCM	3 (predefined)	3 (predefined) Final: 2 (+1 null cluster)	2 (predefined)	6 (predefined) Final: 2 (+ 4 null clusters)
	AP	11	21	36 (+ 9 isolates)	12 (+ 14 isolates)
	DBSCAN	3 (+ 35 isolates)	3 (+ 75 isolates)	2 (+ 194 isolates)	6 (+ 83 isolates)
	Mean Shift	3 (+ 2 isolates)	3 (+ 18 isolates)	2 (+33 isolates + 1 null clusters)	6 (+ 43 isolates)

(c) Comparison results of the next four UCI data sets

Comparison of Clustering Algorithms		Ionosphere	Letter-recognition	Segmentation	Cloud
NMI	SSynC, ESynC	0.3106	0.3986	0.6086	**1**
	SynC	0.3339	**0.5768**	0.6033	0.3016
	K-Means	0.1299	0.3572	0.6103	0.9944
	FCM	0.1264	0.0095	0.4454	0.9944
	AP	0.2809	-	**0.6781**	0.4107
	DBSCAN	**0.4061**	0.1517	0.4592	**1**
	Mean Shift	0.2831	0.3649	0.6447	**1**
AMI	SSynC, ESynC	0.1073	**0.3986**	0.4212	**1**
	SynC	3.5016e-04	0.0166	-1.6974e-15	2.4432e-04
	K-Means	0.1246	0.3484	**0.5286**	0.9944
	FCM	0.1211	0.0042	0.2574	0.9944
	AP	0.1002	-	0.4897	0.1653
	DBSCAN	**0.3417**	0.1517	0.4016	**1**
	Mean Shift	0.0991	0.3649	0.5048	**1**
The Number of Clusters	SSynC, ESynC	2 (+ 83 isolates)	26 (+ 8 isolates)	7 (+ 31 isolates)	2
	SynC	0 (+ 350 isolates)	845 (+ 17823 isolates)	0 (+ 210 isolates)	5 (+ 2038 isolates)
	K-Means	2 (predefined)	26 (predefined)	7 (predefined)	2 (predefined)
	FCM	2 (predefined)	26 (predefined) Final: 2 (+ 24 null clusters)	7 (predefined) Final: 2 (+ 5 null clusters)	2 (predefined)
	AP	14 (+ 44 isolates)	-	17 (+ 7 isolates)	66 (+ 1 isolate)
	DBSCAN	2 (+ 145 isolates)	28 (+ 323 isolates)	7 (+ 51 isolates)	2
	Mean Shift	2 (+ 76 isolates)	26 (+ 3 isolates + 1 null cluster)	7 (+ 22 isolates)	2

Note1: In the Letter-recognition data set, DBSCAN algorithm obtains 21 clusters and 243 isolates when parameter $\delta = 160.0001$, so we set parameter $\delta = 160$ in DBSCAN. The sign '-' in AP column means that the time cost is too larger.

Note2: In Stable 9, the largest values of NMI and AMI in every data set are shown in bold.

4.4 Experimental results of three bmp pictures

Stable 10. Compare three different synchronization algorithms (SynC, ESynC and SSynC) by using three picture data sets. In Stable 10, parameter δ = 18 or 30 in SynC, ESynC and SSynC; parameter ε = 1 in SSynC algorithm.

(a) parameter δ = 18

Comparison of Algorithms		Picture1	Picture2	Picture3
Spend time (second)	SynC	662	676	9795
	ESynC	132	122	3254
	SSynC	**18**	**16**	**297**
Iterative times	SynC	50	50	50
	SSynC, ESynC	**10**	**9**	**16**
The number of steady locations	SynC	941	467	2868
	SSynC, ESynC	**13**	**5**	**14**
The cluster order parameter r_c	SynC	58.6149	118.4821	88.4415
	ESynC	**2712.8392**	**3321.3298**	**6127.5541**
	SSynC	**0**	**0**	**0**
AveLen(T)	SynC	11.0537	10.5757	11.5605
	SSynC, ESynC	**0**	**0**	**0**

(b) parameter δ = 30

Comparison of Algorithms		Picture1	Picture2	Picture3
Spend time (second)	SynC	749	797	10930
	ESynC	122	179	2139
	SSynC	**16**	**16**	**274**
Iterative times	SynC	50	50	50
	SSynC, ESynC	**9**	**13**	**10**
The number of steady locations	SynC	928	472	2896
	SSynC, ESynC	**4**	**2**	**6**
The cluster order parameter r_c	SynC	55.2653	106.8353	87.9900
	ESynC	**3630.5206**	**5015.0178**	**11105.6154**
	SSynC	**0**	**0**	**0**
AveLen(T)	SynC	16.9417	17.5013	19.0378
	SSynC, ESynC	**0**	**0**	**0**

Note: The bold in Stable 10 marks the better results of SSynC algorithm or ESynC algorithm.

Origina Picture SSynC, ESynC (final k = 14) SynC (final k = 2868)

Kmeans, FCM (final k = 1) DBSCAN (final k = 112) Mean Shift (final k = 10)

(a) δ = 18 for SynC, ESynC, SSynC, DBSCAN, and Mean Shift; predefined k (number of clusters) = 14 for KMeans and FCM.

Origina Picture SSynC, ESynC (final k = 6) SynC (final k = 2896)

Kmeans, FCM (final k = 1) DBSCAN (final k = 35) Mean Shift (final k = 4)

(b) δ = 30 for SynC, ESynC, SSynC, DBSCAN, and Mean Shift; predefined k (number of clusters) = 6 for KMeans and FCM.

Sfig. 11. Compare the original picture and several compressed pictures of Picture3 by clustering pixel points of Picture3 in RGB color space using several algorithms. In Sfig. 11, several compressed pictures are drawn using the means of clusters obtained by clustering 200 * 200 pixel points of Picture3 in RGB space.