


<b>ITC 1/49</b> Information Technology and Control Vol. 49 / No. 1 / 2020 pp. 127-143 DOI/10.5755/j01.itc.49.1.23780	<b>Data Cube Clustering with Improved DBSCAN Based on Fuzzy Logic and Genetic Algorithm</b>	
	Received 2019/07/06	Accepted after revision 2019/10/22
	 <a href="http://dx.doi.org//10.5755/j01.itc.49.1.23780">http://dx.doi.org//10.5755/j01.itc.49.1.23780</a>	

**HOW TO CITE:** Rad, M. H., & Abdolrazzagh-Nezhad, M. (2020). Data Cube Clustering with Improved DBSCAN Based on Fuzzy Logic and Genetic Algorithm. *Information Technology and Control*, 49(1), 127-143. <https://doi.org//10.5755/j01.itc.49.1.23780>

# Data Cube Clustering with Improved DBSCAN Based on Fuzzy Logic and Genetic Algorithm

## Mina Hosseini Rad

Department of Computer Engineering, Birjand Branch, Islamic Azad University, Birjand, Iran;  
e-mail: hosseinirad.edu@gmail.com

## Majid Abdolrazzagh-Nezhad

Department of Computer Engineering, Faculty of Engineering, Bozorgmehr University of Qaenat, Qaen, Iran;  
e-mail: abdolrazzagh@buqaen.ac.ir

Corresponding author: abdolrazzagh@buqaen.ac.ir

Multi-dimensional data, such as data cube, are constructed based on aggregating data in data warehouses. Classic pattern recognition methods cannot be applied on the data and it requires to new pattern recognition methods with high flexibility. Moreover, clustering, which is an unsupervised pattern recognition method, has significant challenges to perform on data cube. In this paper, two new drafts of density-based clustering methods are designed to recognize unsupervised patterns of the data cube. In the first draft, DBSCAN clustering is hybridized by genetic algorithm and called the Improved DBSCAN (IDBSCAN). The motivation of designing the IDBSCAN optimizes the DBSCAN's parameters by a meta-heuristic algorithm such as GA. The second draft, which is called the Soft Improved DBSCAN (SIDBSCAN), is designed based on fuzzy tuning parameters of the GA in the IDBSCAN. The fuzzy tuning parameters are performed with two fuzzy groups rules of Mamdani (SIDBSCAN-Mamdani) and Sugeno (SIDBSCAN-Sugeno), separately. These ideas are proposed to present efficient and flexible unsupervised analysis for a data cube by utilizing a meta-heuristic algorithm to optimize DBSCAN's parameters and increasing the efficiency of the idea by applying dynamic tuning parameters of the

algorithm. To evaluate the efficiency, the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno are compared with the IDBSCAN and the DBSCAN. The experimental results, consisted of 20 times running, indicate that the proposed ideas achieved to their targets.

**KEYWORDS:** Data Cube; DBSCAN Clustering; Fuzzy Logic Controller; Dynamic Tuning Parameters; Genetic Algorithm; Meta-Heuristic Algorithm.

---

## 1. Introduction

With regard to the increase an expansion of data on different storage media, there is a natural need for the effective methods for accessing data and extracting useful knowledge. Data mining has been known as one of the most effective methods in this field. Data mining is an iterative process in order to discover knowledge, which is done manually and automatically. Data mining searches for valuable and new information from the huge volume of data [12].

In general, the main aims of data mining include description and prediction. In the first category, data attributes are described in a data set and its focus is about finding patterns from the data set so that the found patterns can be described by human. The second category is based on data deduction, looking for unknown variables and values of the data [26]. Each of these categories includes different patterns such as exploring frequent patterns, classification and regression, clustering and exploring outline patterns, which each of them has its own application and features. The aim of this study is to investigate the clustering analysis which is part of the descriptive patterns with regard to the type of data used for data mining [29].

In clustering, we can create a grouping of data, and so its main aim is to maximize similarity between samples of a cluster as well as minimizing similarity between samples of the various clusters [3]. The clustering widely helps discovery of unknown patterns in data and has a vast application in the various fields, including web searching, security and Business Intelligence as well [13, 35]. Data mining in business intelligence as a powerful and advanced technology will enable companies to have more focus on important data in data warehouses. It can help corporations to effectively adopt Knowledge-based decisions in order to increase business profits using data mining tools [14].

Multi-dimensional data analysis is one of the most important factors in improving efficiency and increasing the data mining speed in business intelli-

gence. In this study, data cube clustering is proposed which this type of data provides the possibility of analysis in various aspects. In the following, some of the works done in the data cube are investigated.

The data warehouses that contain collected data from data sources and are around a specific topic provide possible widespread. The data require the complex analysis for managers by using OLAP tools [14]. The data warehouse and OLAP tools are based on a multi-dimensional data model; therefore, the data cube is the best concept for data modeling in several dimensions, in which data are represented by dimensions and facts. In additional, it is possible to use the OLAP operation in order to create views, interactive query and data analyzing in the data cube [6].

With regard to [23], OLAP is introduced as the main component of business intelligence and data cube is considered as an OLAP's main component. Moreover, it considers the data cube as the most powerful tool for using in Big Databases. The study introduces intelligent cube in order to reduce system response time and also addresses to use compression techniques to reduce storage memory space.

Introducing clustering algorithm for modeling of the data cube and collecting information from cuboid has been already done in [36]. In this study, the amounts of special attributes contain flow of large data and cuboids are used for saving different parts of flow data, and so clustering is carried out on this type of data. Research was done on hierarchical-based clustering algorithm [5] through continuous data and the aim was using the algorithm in applications including wireless sensor network.

In the current research, data cube clustering is considered to prepare an efficient unsupervised analysis through the data. The challenge is the existence of specific and irregular data in the data volume, which cannot be done easily clustering over them. There

are several approaches to clustering [19], which include partitioning method, hierarchical-based clustering, density-based clustering and grid-based clustering and among these four approaches, only a density-based approach has the ability to identify non convex clusters.

Therefore, in order to achieve higher efficiency, we use density-based clustering methods. Among the density-based clustering methods, the method of DBSCAN is widely used in comparison to other methods. The popularity reasons of the DBSCAN are its simplicity to performance and its ability to recognize clusters with different sizes and non-convex shapes [3, 8]. Hence, in the current research, the DBSCAN algorithm is selected for density-based clustering. The DBSCAN is a very good candidate to find non-convex clusters in data space [22]. The challenge of the DBSCAN clustering is the cluster's dependence on its parameters such as the neighborhood radius and the minimum points. These parameters are empirically chosen according to the type of data. Thus, the fine-tune of these parameters has a significant role to identify proper clusters.

There are several literatures which tried to improve DBSCAN. In [32], fuzzy set theory was applied to design fuzzy clustering and improve DBSCAN that the authors called Soft DBSCAN. The Soft DBSCAN was a new fuzzy clustering, which offered appropriate primal degrees for data's membership to express proximities of data entities to the cluster centers. A graph-based index structure method Groups [22] was proposed to improve the performance of DBSCAN on high dimensional dataset that accelerated the neighbor search operations. A new measurement criterion [8] was utilized to obtain a distance which calculated based on the threshold analysis of the nearest neighbor with the total neighbors. In [9], the authors combined the partition technique with DBSCAN. The goal was to obtain the proper input parameters for DBSCAN. However, the effectiveness of this method was not evaluated for datasets with different densities. A combination of Gaussian-Means method with DBSCAN [33] was proposed to improve the determination of DBSCAN parameters. However, Gaussian-Means create circular clusters that are not density-based and do not act very well against dense data as well. The DBSCAN clustering was combined

with Binary Differential Evolution [21] to determine the parameters of the DBSCAN. Recently, many Meta-Heuristic algorithms have been presented to improve clustering on various algorithms for reducing clustering sensitivity to the important parameters of the algorithms [2, 7, 28, 38].

Among them, there is a lack of improvement in the DBSCAN as a density-based clustering with a meta-heuristic algorithm such as Genetic Algorithm (GA). Therefore, in this paper, the GA is considered to identify the best parameters for the DBSCAN. The proposed clustering algorithm, which is called the Improved DBSCAN (IDBSCAN), contains a hybridization of the DBSCAN with the GA on the data cube.

The GA's challenge is tuning mutation and crossover parameters that are also empirically determined. The parameters have a significant impact on the efficiency and convergence of the algorithm. Several adaptive GA's parameters settings were proposed to tune the parameters such as proposing a diversity measure between chromosomes in the population, designing a variation depending on the fitness function values of chromosomes and planning fuzzy logic controllers (FLCs) [1, 20]. An FLC is constructed with a knowledge base, which consists of the information given by linguistic control rules, a fuzzification interface, an inference system and a defuzzification interface [15, 24]. Combining different types of the FLC's elements were created various fuzzy adaptive GA's parameters setting in the previous literatures [15, 16, 24]. Therefore, in order to modify the IDBSCAN, another data cube clustering algorithm is proposed which attempts to improve and accurate the selection of effective GA's parameters by using a new simple FLC based on two groups linguistic control rules such as Mamdani's rules and Takagi-Sugeno's rules. We name the second algorithm as "Soft Improved DBSCAN" (SIDBSCAN). The proposed ideas and their achievements will be considered to design 3D Clustering in our future work.

The rest of the paper is organized as follows. In Section 2, we describe the structures of the proposed algorithms such as preprocessing of the data cube, the DBSCAN, the IDBSCAN, the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno. Then, in the following, the experimental results are illustrated. Finally, conclusion and future works are explained in Section 5.

## 2. The Structures of the Proposed Algorithms

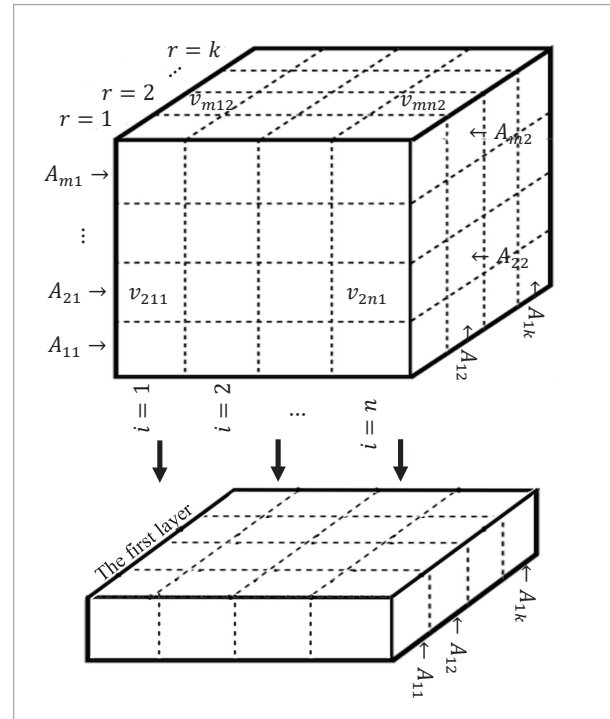
With regard to focusing data cube clustering in the current research, during analysing of the data cube which they may have a special and unusual form, accordingly, we need to use a suitable clustering algorithm to extract appropriate cluster from them rather than the conventional clustering algorithms. To solve this problem, scientists have introduced the density-based clustering. The density-based clustering has introduced clusters, as completely dense areas of samples in comparison with sparse areas. The DBSCAN is one of the most significant methods of the family of clustering methods. In the next subsections, we will introduce preprocessing of the data cube, DBSCAN algorithm as well as the challenge of DBSCAN algorithm and novel strategies to improve it.

### 2.1. Preprocessing of Data Cube

Data is often stored, retrieved and analyzed in a matrix / table structure (two-dimensional with two indexes). A data cube with its three-dimensional structure, which requires three indexes to storage, makes challenge to define and implement data mining techniques. The structure of the 3D data cube is the research data structure and 3D data cube density-base clustering is the current research problem. Hence, a 3D data cube is shown with three storage indexes in Fig. 1. In the previous literatures, one dimension of the 3D data cube was ignored in preprocessing, then clustering techniques were performed over it [5, 30, 36, 37]. Considering information of the 3D data cube without any deletion is the main aim of the current research. To achieve to the aim, a one-to-one linear transfer function is proposed to transfer the information of the third dimension into the 2D space along one dimension of the space. It points that the transfer function has reversible capacity and the obtained results can be returned in the original 3D space. Therefore, the 2D results are interpretable to the 3D space. In Fig. 1, a 2D information shows by extracting a slice from the data cube. The 2D information can be connected together along one of their dimensions.

To perform data cube clustering, two main preprocessing steps should be passed such as normalizing data cube and converting three-dimensional of the normalized data into two-dimensional data. There

Figure 1  
Indexing a Data Cube



are different scaling sizes between the 3D attributes of the data and application of normalizing the attributes which is caused by removing the effect of larger scale attributes on smaller one. The min-max normalization [13] is a linear converter of  $v_i \in A$  into  $v'_i \in A'$  with new bound  $[new\_min_{A'}, new\_max_{A'}]$ . The min-max normalization was proposed for 2D data and a 3D draft of the normalization (see Fig. 1), which is linearly converted to  $[0, 1]$ , is designed in the current research that its procedure is shown for the first layer as follows:

$$\left\{ \begin{array}{l} v'_{1i1} = \frac{v_{1i1} - \min_{1 \leq j \leq n}(A_{11})}{\text{Max}_{1 \leq j \leq n}(A_{11}) - \min_{1 \leq j \leq n}(A_{11})}, i = 1, \dots, n \\ \vdots \\ v'_{1ik} = \frac{v_{1ik} - \min_{1 \leq j \leq n}(A_{1k})}{\text{Max}_{1 \leq j \leq n}(A_{1k}) - \min_{1 \leq j \leq n}(A_{1k})}, i = 1, \dots, n \end{array} \right. \quad (1)$$

The normalized values of the next layers are calculated based on Equation(1). Because data cube consists of 3D, it would be better to convert into 2D matrices. There are technical reshape data cube and 3D matrix such as [18, 31, 39] that have different complexity and are applied in

image processing. In the current research, we use a simple reshape method which is accessible in MATLAB. A data cube called X, which has dimensions  $m \times n \times k$ , can be converted to a 2D Y matrix with dimension  $m \cdot k \times n$  with the following reshape command:

$$Y = \text{reshape}(X, [m \times k, n]). \quad (2)$$

## 2.2. DBSCAN Algorithm

DBSCAN [13] is an information clustering method based on the data density that its brief procedure is presented in Algorithm 1. Two parameters such as the neighborhood radius ( $\epsilon$ ) and minimum points (MinPts) ( $\mu$ ) are needed to form a cluster have been used in order to evaluate the distributed density of points. This algorithm begins from an optional point and then it accounts the points which are located in the neighborhood radius of this point at a distance less than  $\epsilon$ . If the number of points is more than  $\mu$  parameter, they produce a cluster; otherwise, the intended point is known as an outlier data. In the next step, this point may be recognized as a part of a cluster. The advantage of this method is the ability to distinguish and separate the outlier data from other data.

To evaluate the obtained clusters with DBSCAN, the Davies Bouldin Index (DBI) [10] is considered. It calculates within-cluster's distance and between clusters distance. The best choice of clusters will be done, since the DBI is minimized and the index is formulated as follows:

$$DBI = \frac{1}{N} \sum_{i=1}^N \left( \max_{\substack{j=1, \dots, N \\ j \neq i}} \left( \frac{S_i + S_j}{d_{ij}} \right) \right), \quad (3)$$

### Algorithm 1 : DBSCAN Clustering

**Input:** N objects to be clustered, the neighborhood radius ( $\epsilon$ ) and minimum points ( $\mu$ )

- 1: Randomly select a point  $P$
- 2: Retrieve all points density-reachable from  $P$  based on  $\epsilon$  and  $\mu$
- 3: If  $P$  is a core point, a cluster is formed.
- 4: If  $P$  is a border point, no points are density-reachable from  $P$  and DBSCAN selects the next no-visited point randomly.
- 5: Continue the procedure until all points have been processed.

where  $N$  is the number of clusters,  $d_{ij}$  is the average linkage as between-cluster's distance of clusters  $C_i$  and  $C_j$ ,  $S_i$  and  $S_j$  are the average distance of within-cluster  $C_i$  and within-cluster  $C_j$ , respectively.

$$d_{ij} = \frac{\sum_{p_r \in C_i, p_s \in C_j} \|p_r - p_s\|}{\|C_i\| \times \|C_j\|}, \quad (4)$$

$$S_i = \frac{\sum_{p_r, p_s \in C_i} \|p_r - p_s\|}{\|C_i\| (\|C_i\| - 1)}, \quad (5)$$

where  $\|\bullet\|$  is Euclidean norm and  $p_r \in C_i$  means point  $r$  belong to the cluster  $i$ .

In this algorithm, the most important role is to find the proper  $\epsilon$  and  $\mu$  points. Commonly, using statistical and classical methods of combining different data mining ways can find these points. In many cases, despite consuming too much time, this is not run with high precision. Therefore, in the research, we try to use the Genetic Algorithm (GA), as a meta-heuristic algorithm, to estimate the exact values for these parameters and achieve significant improvements.

## 2.3. The Improved DBSCAN

To design the improved DBSCAN (IDBSCAN), GA is adapted to find the optimum values for  $P$ ,  $\mu$  and  $\epsilon$  in Algorithm 1. In the adapted GA to improve DBSCAN for a dataset with  $N$  objects/points and  $M$  attributes, each chromosome is an  $M+2$  dimensional array such as Equation (6). The first  $M$  elements represent an initial point  $P$  that DBSCAN starts with. The element of  $M+1$  represents the neighboring radius ( $\epsilon$ ) and the last element represents the value of MinPts ( $\mu$ ).

$$\text{Chro}_i = \begin{cases} 0 \leq \text{chor}_{ij} \leq 1, & \text{if } 1 \leq j \leq M \\ \text{dis}_{\min} \leq \text{chor}_{ij} \leq \text{dis}_{\max}, & \text{if } j = M + 1 \\ 2 \leq \text{chor}_{ij}, & \text{if } j = M + 2 \end{cases} \quad (6)$$

$$\text{dis}_{\min} = \min_{\substack{1 \leq i, r \leq N \\ i \neq r}} \|p_i, p_r\|, \quad (7)$$

$$\text{dis}_{\max} = \max_{\substack{1 \leq i, r \leq N \\ i \neq r}} \|p_i, p_r\|, \quad (8)$$

where,  $i, r = 1, \dots, \text{pop\_size}$ ,  $\text{dis}_{\min}$  and  $\text{dis}_{\max}$  are the minimum and the maximum distances between objects/points. The GA's inputs are population size



( $pop\_size$ ), crossover rate ( $P_c$ ), mutation rate ( $P_m$ ), maximum iterations ( $MaxItr$ ) and/or other terminate criteria [4, 11, 17, 27]. In Algorithm 2, a brief procedure of the GA is presented based on initialization, crossover, mutation and selection. With regard to the chromosome's structure in Equation (6),  $pop\_size$  chromosomes are generated, as initial population in the zero generation  $P(0)$ , randomly. To evaluate each chromosome, Algorithm 1 runs and the DBI (3) calculates as its fitness function.

Algorithm 2 consists of scattered crossover (See Fig. 2) based on  $R_c$  that  $Child_1$  and  $Child_2$  are generated by combining  $Par_1$  and  $Par_2$  with regard a random binary array ( $Ran\_Bin$ ). Roulette cycle is considered to select new population  $P(t+1)$  from  $P(t) \cup C(t)$  as the selection rule. Optimal determination of the mutation and the crossover rates is the GA's chal-

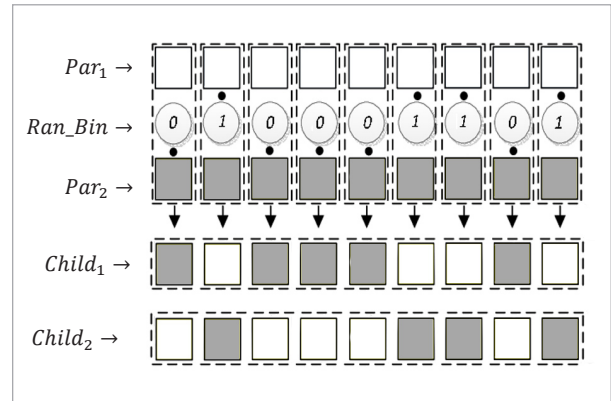
#### Algorithm 2: Adapted GA

**Input:**  $pop\_size, P_c, P_m$  and  $MaxItr$

- 1:  $t = 0$
- 2: Initialization: Generate  $pop\_size$  chromosomes based on (6) randomly as the initial population  $P(t)$ .
- 3: Evaluate  $P(t)$  with run Algorithm1 and calculate (3).
- 4: Get the best of  $P(t)$  as  $P_{best}$
- 5: while ( $t \leq MaxItr$ ) do
- 6: Select Parents based on  $P(t)$  and  $pop\_size$ .
- 7: Get an empty set of children  $C(t)$ .
- 8: for each ( $Par_1, Par_2 \in Parents$ ) do
- 9: Generate  $Child_1$  and  $Child_2$  based on  $P_c$  and crossover of  $Par_1, Par_2$ .
- 10: Mutate  $Child_1$  based on  $P_m$  and save it in  $C(t)$ .
- 11: Mutate  $Child_2$  based on  $P_m$  and save it in  $C(t)$ .
- 12: end.
- 13: Evaluate  $C(t)$  by Algorithm1 and calculate (3).
- 14: Select new population  $P(t+1)$  based on  $P(t) \cup C(t)$  and a given selection rule.
- 15: Get the best of  $P(t+1)$  as  $P_{best}$ .
- 16:  $t = t + 1$
- 17: end

Figure 2

Scattered Crossover Procedure



lenge. These parameters are empirically determined and have a significant impact on the efficiency, accuracy and speed up of the algorithm.

#### 2.4. The Soft Improved DBSCAN Algorithm

To fill up the above challenge and to design the soft improved DBSCAN (SIDBSCAN), a self-adaptive GA based on a new fuzzy logic controller (FLC) is designed in this subsection.

Algorithm 3 has the procedure like Algorithm 2, except calculating  $P_c$  and  $P_m$  by the proposed FLC (See Fig. 3) based on the inputs such as the  $UN$  and the  $fitBest$ . The FLC's inputs are normalized values related to iteration and the evaluation function (3) that they calculate as follows:

$$UN = \frac{t}{MaxItr}, \quad (9)$$

$$fitBest = \frac{DBI(P_{best})}{\min(DBI(Par_1), DBI(Par_2))}. \quad (10)$$

In Equations (9)-(10),  $UN$  specifies the number of iterations where the best fitness value is not improved in  $P(t)$ ,  $t$  is the number of iteration and  $MaxItr$  is the number of the maximum iterations.  $P_{best}$  represents the best chromosome in  $P(t)$ . In the FLC system, fuzzy rules are specified based on diversity, linguistic values of the input variables. Since each input variable consists of three fuzzy linguistic values such as high, medium and low in the system, nine ( $3 \times 3$ ) fuzzy rules could be written that we extracted just five rules from them. The cause of ignoring four

**Algorithm 3: Self-Adaptive GA****Input:**  $pop\_size, P_c, P_m$  and  $MaxItr$ 

```

1:  $t = 0$ 
2: Initialization: Generate  $pop\_size$  chromosomes based on (6) randomly as the initial population  $P(t)$ .
3: Evaluate  $P(t)$  with run Algorithm1 and calculate (3).
4: Get the best of  $P(t)$  as  $P_{best}$ 
5: while ( $t \leq MaxItr$ ) do
6:   Select Parents based on  $P(t)$  and  $pop\_size$ .
7:   Get an empty set of children  $C(t)$ .
8:   Calculate UN and fitBest by (9) and (10) respectively
9:   Calculate  $P_c$  and  $P_m$  by FLC ( $UN, fitBest$ ) (Fig. 3)
10:  for each ( $Par_1, Par_2 \in Parents$ ) do
11:    Generate  $Child_1$  and  $Child_2$  based on  $P_c$  and crossover of  $Par_1, Par_2$ .
12:    Mutate  $Child_1$  based on  $P_m$  and save it in  $C(t)$ .
13:    Mutate  $Child_2$  based on  $P_m$  and save it in  $C(t)$ .
14:  end.
15: Evaluate  $C(t)$  by Algorithm1 and calculate (3).
16: Select new population  $P(t+1)$  based on  $P(t) \cup C(t)$  and a given selection rule.
17: Get the best of  $P(t+1)$  as  $P_{best}$ .
18:  $t = t + 1$ 
19: end

```

other rules is difficult to define effective fuzzy linguistic values of their output variables. The membership functions of low, medium and high are Z-Shape, Gaussian and S-Shape with parameters [0,0.5], [0.12,0.5] and [0.5,0.99], respectively. The extracted fuzzy rules of the FLC are listed as follows:

*Rule 1: If (UN is HIGH) and (FitBest is HIGH) then*  
 $(P_m \text{ is HIGH})(P_c \text{ is High})$

*Rule 2: If (UN is LOW) and (FitBest is LOW) then*  
 $(P_m \text{ is LOW})(P_c \text{ is LOW})$

*Rule 3: If (UN is MEDIUM) and (FitBest is LOW) then*  
 $(P_m \text{ is MEDIUM})(P_c \text{ is MEDIUM})$

*Rule 4: If (UN is MEDIUM) and (FitBest is MEDIUM) then*  
 $(P_m \text{ is HIGH})(P_c \text{ is MEDIUM})$

*Rule 5: If (UN is HIGH) and (FitBest is LOW) then*  
 $(P_m \text{ is LOW})(P_c \text{ is MEDIUM})$ .

*Rule 1* states that if the value of the fitness function is far from optimal and there are many iterations that do not improve, the mutation rate and crossover rate should be highly selective to improve population variation and improve the fitness function. *Rule 2* also shows that if the value of the fitness function is close to optimal and improves at almost every iteration, the value of the mutation and crossover rate should be low selected so that the good genes on the chromosomes do not destroy. *Rule 3* presents when the value of the fitness function is close to optimal, but has several iterations that has not improved. In order to improve the value of the fitness function, the rate of mutation and crossover can be set to average. *Rule 4* displays that if the value of the fitness function is moderate and it is repeated that this value is not improved, a high mutation rate can be selected to cause chromosome variation and the average crossover rate value retains reproducibility. *Rule 5* expresses that there are many iterations where the fitness value is close to optimal. In this case, the reproducibility of the average crossover rate should be maintained, but the algorithm's scalability is reduced to the mutation rate.

The difference between the Mamdani and Sugeno's fuzzy systems, as stated in the fuzzy rules, is given below in the rules of the Sugeno fuzzy system to improve mutation rates and crossover GA. The outputs variables of the Sugeno's rules are calculated based on given crisp linear hyper-lines. Since there are two input variables,  $UN$  and  $FitBest$ , the linear hyper-lines consists of two variables. Defining appropriate coefficients of the Sugeno's rules is its main challenge. Thus, five fuzzy Sugeno rules are extracted from nine possible rules as follows:

*Rule 1: If (UN is HIGH) and (FitBest is HIGH) then*

$$(P_m = 0.5 \times UN + 0.5 \times FitBest + 0.1)$$

$$(P_c = 0.3 \times UN + 0.3 \times FitBest + 0.1)$$

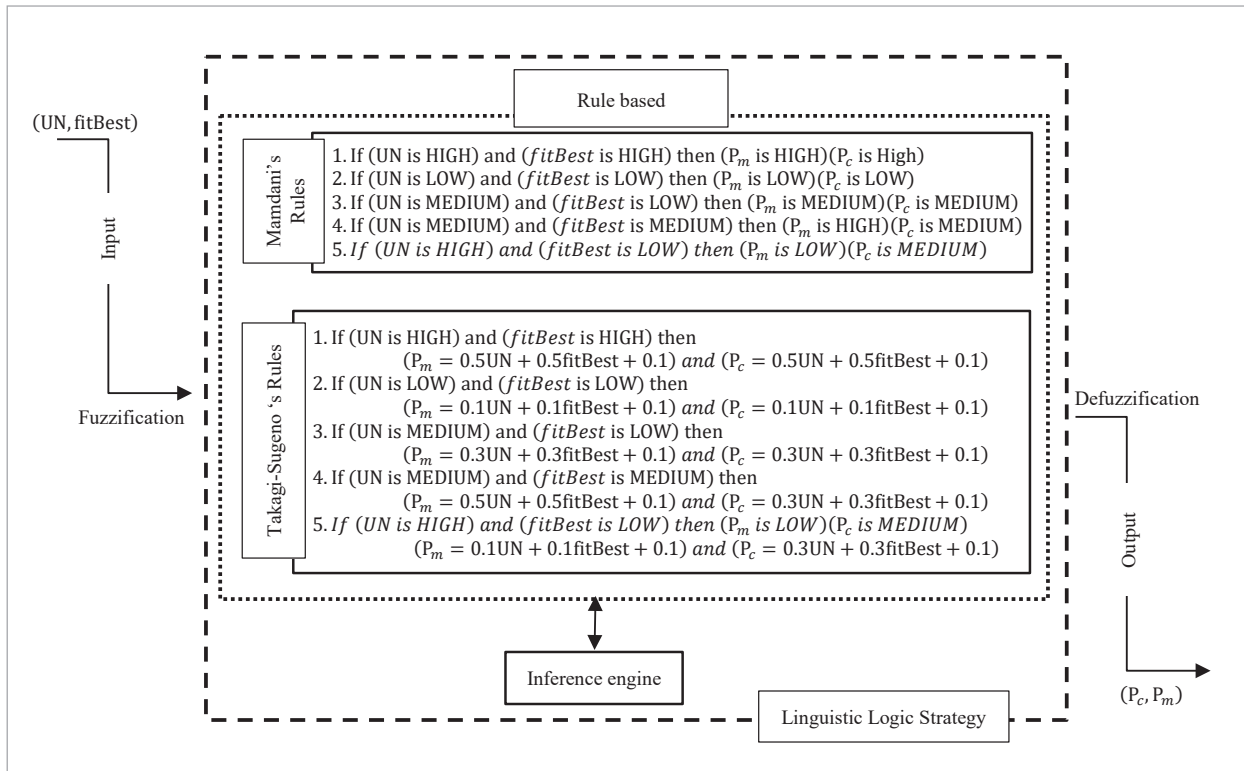
*Rule 2: If (UN is LOW) and (FitBest is LOW) then*

$$(P_m = 0.1 \times UN + 0.1 \times FitBest + 0.1)$$

$$(P_c = 0.1 \times UN + 0.1 \times FitBest + 0.1)$$

Figure 3

The proposed fuzzy logic controller based on two inputs (UN,fitBest) and two outputs ( $P_c, P_m$ )



Rule 3 : If (UN is MEDIUM) and (FitBest is LOW) then

$$(P_m = 0.4 \times UN + 0.4 \times FitBest + 0.1)$$

$$(P_c = 0.3 \times UN + 0.3 \times FitBest + 0.1)$$

Rule 4 : If (UN is MEDIUM) and (FitBest is MEDIUM) then

$$(P_m = 0.5 \times UN + 0.5 \times FitBest + 0.1)$$

$$(P_c = 0.3 \times UN + 0.3 \times FitBest + 0.1)$$

Rule 5 : If (UN is HIGH) and (FitBest is LOW) then

$$(P_m = 0.1 \times UN + 0.1 \times FitBest + 0.1)$$

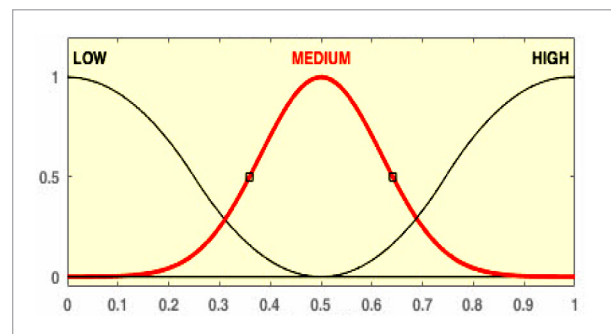
$$(P_c = 0.3 \times UN + 0.3 \times FitBest + 0.1)$$

The FLC consists of fuzzifying the inputs, linguistic logic strategy (LLS) and defuzzifying the outputs. The inputs fuzzify based on the presented membership functions in Fig. 4. The LLS includes two main parts, naming, rule based and inference engine. There are two groups rules, which are called Mamdani's rules and Takagi-Sugeno's rules, because the FLC is designed to generate dynamic outputs based on Mamdani's [25] and Takagi-Sugeno's inferences [34]. The LLS are cre-

ated outputs' surfaces in Figures 5-6 by Mamdani's rules and Takagi-Sugeno's rules, respectively. Maximum and minimum operations are considered for "OR" and "AND" operators in the LLS's inference engine to aggregation functions and reasoning. The center of gravity is used for defuzzification method.

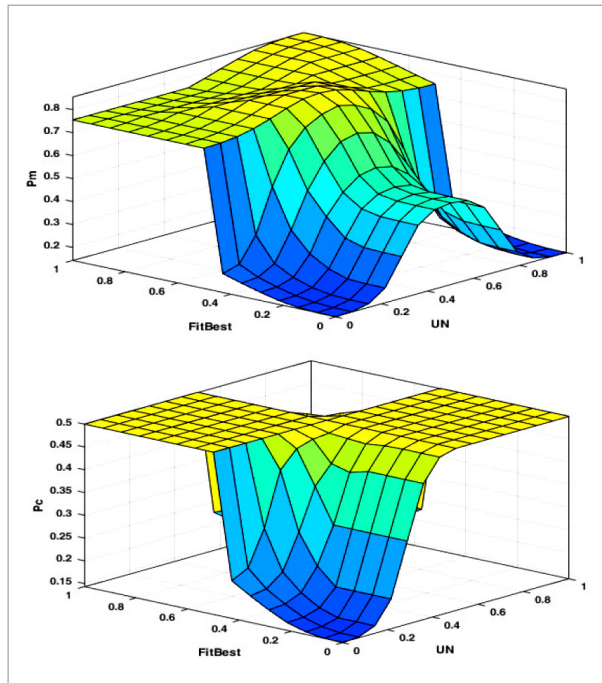
Figure 4

The membership functions of two inputs (UN,fitBest) based on linguistic values of low, medium and high

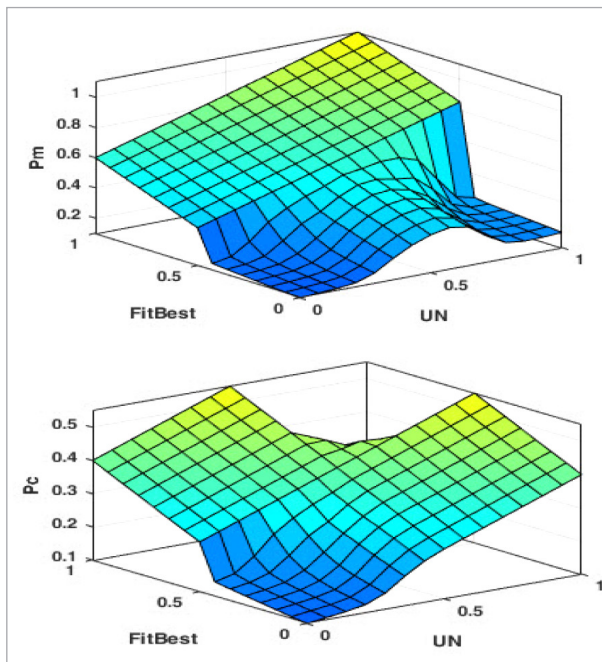




**Figure 5**  
The outputs' surfaces of the LLS based on Mamdani's rules



**Figure 6**  
The outputs' surfaces of the LLS based on Takagi-Sugeno's rules



### 3. Evaluation of the Improved DBSCAN Algorithm and the Soft Improved DBSCAN

To evaluate the effectiveness of the proposed algorithms, experiments were performed on the Intel Core i5 3.2 GHz CPU and 4.00 GB memory. The algorithms were implemented in Matlab 2017a. Six benchmark datasets of the data cube, which are available from UCI, and are considered for experimentation, are shown in Table 1.

**Table 1**  
The investigated data cube

ID	Dataset cube	Dimensions
1	Daily Demand Forecasting Orders	$8 \times 12 \times 5$
2	Istanbul Stock Exchange	$20 \times 9 \times 26$
3	Dow Jones Index	$330 \times 14 \times 2$
4	ADL Recognition	$844 \times 3 \times 10$
5	Software Engineering Teamwork	$63 \times 84 \times 5$
6	User Identification From Walking Activity	$1144 \times 4 \times 6$

There are five parameters in the experimentation, such as  $\mu$  and  $\epsilon$  in *Algorithm 1*,  $pop\_size$ ,  $P_c$ ,  $P_m$  and  $MaxItr$  in *Algorithm 2* and  $pop\_size$  and  $MaxItr$  in *Algorithm 3*. Tuning parameters of *Algorithm 1* are based on  $\epsilon=0.5$  and  $\mu$  is 10% of the investigated data, because other values increased the DBI and number of clusters simultaneously. *Algorithm 2* was tested on the data cube of "Daily Demand Forecasting Orders" by different values for  $P_c$  and  $P_m$ , then  $P_c = 0.8$  and  $P_m = 0.02$ , which had the best results, were considered to experimentations of all data cube. Finally,  $pop\_size$  and  $MaxItr$  were tuned with 100 chromosomes and 100 iterations.

The IDBSCAN and the SIDBSCAN are run 20 times on each data cube, then the best obtained DBI (3) is reported as the best quality clustering of the data. The details of the obtained results, such as NC (the number of obtained clusters) and DBI (Davies Bouldin Index), from implementations of the DBSCAN, the

IDBSCAN, the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno are shown in Appendix A, and Tables 5–8, respectively. These Tables summarized based the best results in Table 2. Comparing the results shows that the IDBSCAN success to improve the quality of data cube clustering between 4% for “User Identification from Walking Activity” until 28% for “Dow Jones Index”. This comparison is calculated for DBSCAN vs IDBSCAN, DBSCAN vs SIDBSCAN-Mamdani and DBSCAN vs SIDBSCAN-Sugeno in Table 3. With regard to the tables of Appendix A and Table 3, the performance of the SIDBSCAN-Sugeno (*Algorithm 3 based on Takagi-Sugeno’s rules*) is significantly superior to that the other performed algorithms. Its causes are dynamic appropriate tuning  $P_c$  and  $P_m$  compared with IDBSCAN and SIDBSCAN-Mamdani and optimal determining the neighborhood radius ( $\epsilon$ ) and minimum points (MinPts) ( $\mu$ ) parameters instead their random values in the DBSCAN. In the competition between SIDBSCAN-Mamdani vs SIDBSCAN-Sugeno, five fuzzy Sugeno rules have achieved better results than five fuzzy Mamadani rules. Because the Sugeno and Mamadani rules could be adjusted a lot in their linguistic values and membership functions of the input and output variables, the superiority of SIDBSCAN-Sugeno over SIDBSCAN-Mamdani is uncertain, but the superiority is certain over DBSCAN and IDBSCAN.

Let the converted 2D data from its related 3D data cube consist of  $N$  objects, then the time complexities of DBSCAN, the GA and the IDBSCAN are  $O(N^2)$ ,  $O(pop\_size)$  and  $O(N^2 pop\_size)$ , respectively. The main difference between SIDBSCAN-Mamdani

**Table 2**

The best results (DBI) after 20 runs

Datasets	DBSCAN	IDBSCAN	SIDBSCAN-Mamdani	SIDBSCAN-Sugeno
1	1.0262	0.7777	0.7279	0.7160
2	0.9610	0.7970	0.7732	0.7676
3	0.7155	0.5120	0.4890	0.4604
4	0.8466	0.8010	0.7977	0.7654
5	0.6418	0.5723	0.5549	0.5520
6	0.6117	0.5862	0.5245	0.5201

**Table 3**

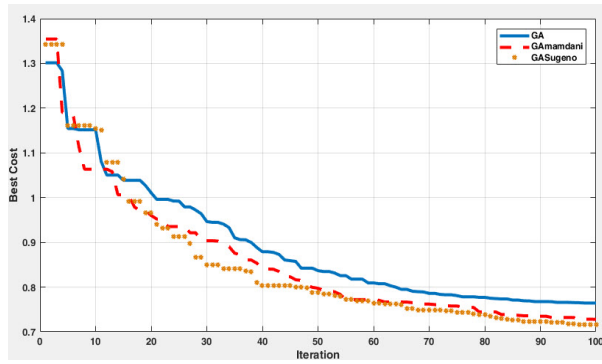
The improvement rates of the proposed algorithms with regard Algorithm 1 for the quality of data cube clustering

Datasets	IDBSCAN	SIDBSCAN-Mamdani	SIDBSCAN-Sugeno
1	24.2 %	29.1 %	30.2 %
2	17.1 %	19.5 %	20.1 %
3	28.4 %	31.7 %	35.7 %
4	5.4 %	5.8 %	9.6 %
5	10.8 %	13.5 %	14.0 %
6	4.2 %	14.3 %	15.0 %

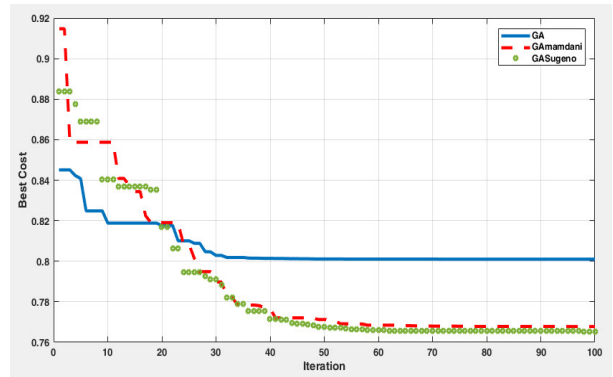
and SIDBSCAN-Sugeno resides in how generated crisp outputs from the fuzzy inputs. While SIDBSCAN-Mamdani utilizes defuzzification of its fuzzy outputs, SIDBSCAN-Sugeno uses a weighted average to compute its crisp outputs, so the SIDBSCAN-Sugeno’s outputs membership functions are linear but SIDBSCAN-Mamdani’s inference expects its output membership function to be fuzzy sets. Therefore, the SIDBSCAN-Sugeno has better processing time, since the weighted average replaces the time consuming defuzzification. As seen in designing *IDBSCAN*, SIDBSCAN-Mamdani and SIDBSCAN-Sugeno, for which they used the GA as a meta-heuristic algorithm, it has been succeeded to design an efficient DBSCAN as a non-convex data cube clustering, but the idea increased run time more than the DBSCAN, which has not been aim of the study.

To compare the functionality of the proposed data cube clustering algorithms, the curves convergences of the best DBI are shown on the datasets in Figures 7–12. The horizontal axis of the figures is measured based on the number of iterations from 1 to 100 and the vertical axis are denoted by the best found DBI thorough improvement. The blue, red and orange lines belong to the improvement curves of the IDBSCAN, the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno, which are denoted with GA, GAMamdani and GASugeno, respectively. Based on the figures, two data cube clustering algorithms from the SIDBSCAN have better improvement and convergence than the IDBSCAN. In Fig. 10 and Fig. 12, the IDBSCAN dropped in local optimal from iteration 30, while the SIDB-

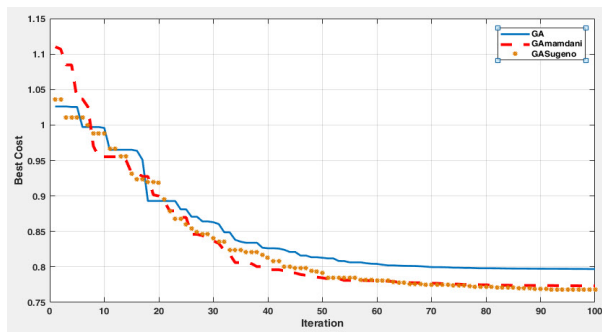
**Figure 7**  
The convergence curves for the Daily Demand Forecasting Orders



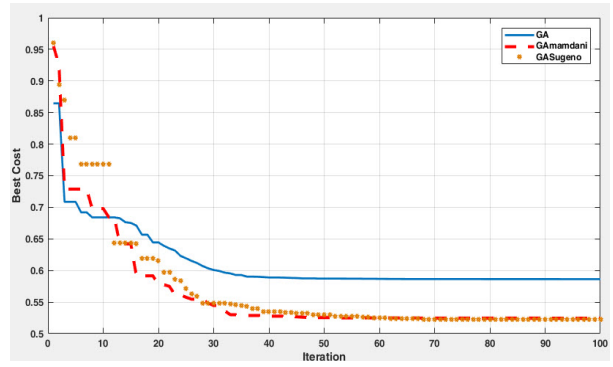
**Figure 10**  
The convergence curves for the ADL Recognition



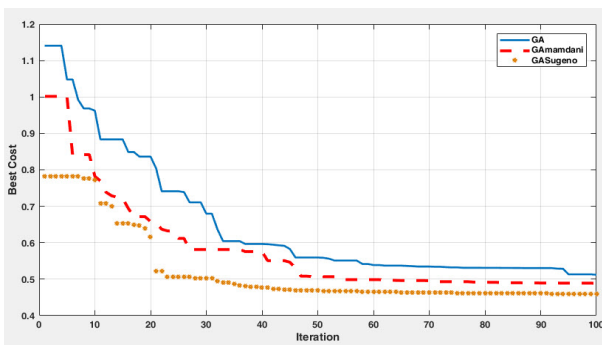
**Figure 8**  
The convergence curves for the Istanbul Stock Exchange



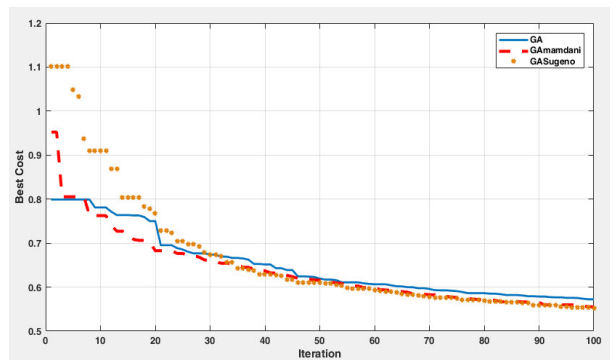
**Figure 11**  
The convergence curves for the User Identification from Walking Activity



**Figure 9**  
The convergence curves for the Dow Jones Index



**Figure 12**  
The convergence curves for the Software Engineering Teamwork



**Table 4**

The results of the Wilcoxon signed rank test in the form of [Z,P]

	DBSCAN vs IDBSCAN	DBSCAN vs SIDBSCAN-Mamdani	DBSCAN vs SIDBSCAN-Sugeno	IDBSCAN vs SIDBSCAN-Mamdani	IDBSCAN vs SIDBSCAN-Sugeno	SIDBSCAN-Mamdani vs SIDBSCAN-Sugeno
1	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-2.427, <b>0.015</b> ]	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]
2	[-3.920, <b>0.000</b> ]	[-3.883, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-1.568, 0.117]	[-3.920, <b>0.000</b> ]	[-3.883, <b>0.000</b> ]
3	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-1.456, 0.145]	[-3.920, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]
4	[-0.821, 0.411]	[-2.501, <b>0.012</b> ]	[-3.969, <b>0.000</b> ]	[-1.232, 0.218]	[-0.821, 0.411]	[-2.501, <b>0.012</b> ]
5	[-3.584, <b>0.000</b> ]	[-3.584, <b>0.000</b> ]	[-3.920, <b>0.000</b> ]	[-1.904, 0.057]	[-3.584, <b>0.000</b> ]	[-3.584, <b>0.000</b> ]
6	[-0.149, 0.881]	[-3.211, <b>0.001</b> ]	[-3.435, <b>0.001</b> ]	[-2.613, <b>0.009</b> ]	[-0.149, 0.881]	[-3.211, <b>0.001</b> ]

SCAN-Mamdani and the SIDBSCAN-Sugeno could improve their qualities of clustering. The IDBSCAN have been begun its improvement data cube clustering better than two algorithms of the SIDBSCAN in Figures 7-12 (except Fig. 9), but the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno converged to better DBIs than the final achievement of the IDBSCAN algorithms from the SIDBSCAN have better improvement and convergence than the IDBSCAN. In Fig.10 and Fig. 12, the IDBSCAN dropped in local optimal from iteration 30, while the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno could improve their qualities of clustering. The IDBSCAN have been begun its improvement data cube clustering better than two algorithms of the SIDBSCAN in Figures 7-12 (except Fig. 9), but the SIDBSCAN-Mamdani and the SIDBSCAN-Sugeno converged to better DBIs than the final achievement of the IDBSCAN.

According to the above convergence curves, the GA performs poorer than the GA-Mamdani and the GA-Sugeno, which utilized the FLC to dynamical tune  $P_c$  and  $P_m$ . Its reason is using fixed values of crossover and mutation rates in the GA, while increasing  $P_m$  vs decreasing  $P_c$  can lead to the exit the GA from trapping in a local optimum and conversely tuning  $P_c$  and  $P_m$  can help to discover appropriate solutions. In addition, a comparison of the GA-Mamdani and the GA-Sugeno shows that the GA-Sugeno has often performed better. To evaluate the significance level of the comparisons for the proposed data cube clustering algorithms, a hypothesis test is done to test the difference in the resulting quality between the algorithms. Because the obtained results of each algorithm are not normally distributed, a non-normal distributed hypoth-

esis test, such as the Wilcoxon Signed-Rank test, is utilized in SPSS between two samples at a significant level of  $\alpha=0.05$ . The results are presented in the form of [Z, P] in Table 4. If P-Value  $<0.05$ , then the null hypothesis (the two samples are dependent samples) can be rejected at the 95% level, but if P-Value  $>0.05$ , then the null hypothesis cannot be rejected. Therefore, the bold P-values present that, the comparison of two mentioned clustering algorithms on the related datasets is significant at the 95% level.

## 4. Conclusion and Future Work

This paper focuses on the data cube density-based clustering. The DBSCAN clustering is considered as the basic clustering technique to apply for data cube clustering. To improve the efficiency of the DBSCAN, two efficient meta-heuristic clustering algorithms, such as the Improved DBSCAN and the Soft Improved DBSCAN, were introduced for data cube clustering. To achieve this aim, we designed a hybridization of the Genetic Algorithm and DBSCAN algorithm to find the optimum values for  $P$ ,  $\mu$  and  $\epsilon$ . The experimental results showed that the proposed algorithm has better quality of clustering than the DBSCAN. However, the IDBSCAN has two challenges to determine optimal values for the mutation and the crossover rates.

To fill up the IDBSCAN's challenges, the soft IDBSCAN algorithms were introduced and called the SIDBSCAN. The algorithms try to tune the mutation and the crossover rates using a fuzzy logic controller (FLC) and enhances the exploration and exploration capabilities of the IDBSCAN. The designed FLC has been carried out by Mamdani's rules and Takagi-Sugeno's rules.

To evaluate and compare the proposed clustering algorithms, six datasets of data cube were considered and the details of the obtained results were reported in Appendix A. All experiments indicated the efficiency and improvement of the SIDBSCAN-Sugeno, although the IDBSCAN and the SIDBSCAN-Mamdani succeeded to improve the quality of the DBSCAN clustering.

Finally, although application of the meta-heuristic algorithm has been succeeded to design an efficient non-convex data cube clustering, but the idea increased run time more than the DBSCAN, which has not been THE aim of the study. It is promising that we can reduce the run time of the algorithms using parallel and distributed processors in the future. Three

dimensions of data cube are reduced into two dimensions in this research, but the achievements of the research will be considered to design novel 3D clustering, which has application in 3D image processing, as our future research object.

## 5. Compliance with Ethical Standards

The study is not funded by any agency. The authors do hereby declare that there is no conflict of interests of other works regarding the publication of this paper. The manuscript does not contain any studies with human participants or animals performed by any of the authors.

## Appendix A

**Table 5**

The details of the experimental results for 20 runs of the DBSCAN (Algorithm 1)

	Daily Demand Forecasting Orders		Istanbul Stock Exchange		Dow Jones Index		ADL Recognition		Software Engineering Teamwork		User Identification From Walking Activity	
	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**
1	1.0776	3	1.0528	3	0.7730	3	0.8672	3	0.7592	4	0.6292	4
2	<b>1.0262</b>	<b>3</b>	1.1977	4	0.8301	3	<b>0.8466</b>	<b>3</b>	0.6852	3	0.6853	4
3	1.0239	3	1.1589	4	0.8918	4	0.9471	4	0.6964	3	<b>0.6117</b>	<b>3</b>
4	1.2061	4	1.0006	3	<b>0.7155</b>	<b>2</b>	0.8639	3	0.7537	4	0.6703	4
5	1.1385	3	<b>0.9610</b>	<b>3</b>	0.8017	3	0.9257	3	0.7299	4	0.6326	3
6	1.1123	3	1.1252	3	0.6985	3	0.8863	3	0.7327	4	0.7943	4
7	1.1833	3	1.0124	3	0.7280	2	0.9833	4	0.7650	4	0.6788	3
8	1.2829	3	1.0595	3	0.8011	3	0.8868	4	0.7383	4	0.6874	3
9	1.2433	3	1.0501	3	0.8571	3	0.9303	4	0.6823	3	0.6575	3
10	1.2922	4	0.9971	2	0.8751	3	0.9115	4	0.6654	2	0.6269	3
11	1.0393	3	1.1844	4	0.7895	3	0.8720	3	0.7289	4	0.6975	3
12	1.0551	3	1.1529	4	0.8025	3	0.9771	4	0.7188	3	0.6596	3
13	1.1059	3	1.0482	3	0.8560	4	0.8625	4	0.7877	4	0.7537	4
14	1.2973	4	1.1419	4	0.8272	3	0.8771	3	0.8123	4	0.6260	3
15	1.2534	4	1.1095	3	0.8021	3	0.9849	4	0.8853	4	0.6971	3
16	1.1245	3	1.1616	4	0.8143	3	0.9947	4	0.7322	4	0.7277	4
17	1.2078	3	0.9770	3	0.7337	3	0.9437	4	0.7881	4	0.6536	3
18	1.0307	4	1.0180	2	0.8166	3	0.8598	3	0.8615	4	0.6998	3
19	1.2267	3	1.0835	2	0.7956	3	0.9644	4	0.8718	4	0.6473	3
20	1.2367	4	1.1481	4	0.8185	3	0.9735	4	<b>0.6418</b>	<b>2</b>	0.7238	4

\*Davis Boulder Index (DBI), \*\*Number of Clusters



**Table 6**  
The details of the experimental results for 20 runs of the DBSCAN (Algorithm 1)

	Daily Demand Forecasting Orders		Istanbul Stock Exchange		Dow Jones Index		ADL Recognition		Software Engineering Teamwork		User Identification From Walking Activity	
	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**
1	1.0776	3	1.0528	3	0.7730	3	0.8672	3	0.7592	4	0.6292	4
2	<b>1.0262</b>	<b>3</b>	1.1977	4	0.8301	3	<b>0.8466</b>	<b>3</b>	0.6852	3	0.6853	4
3	1.0239	3	1.1589	4	0.8918	4	0.9471	4	0.6964	3	<b>0.6117</b>	<b>3</b>
4	1.2061	4	1.0006	3	<b>0.7155</b>	<b>2</b>	0.8639	3	0.7537	4	0.6703	4
5	1.1385	3	<b>0.9610</b>	<b>3</b>	0.8017	3	0.9257	3	0.7299	4	0.6326	3
6	1.1123	3	1.1252	3	0.6985	3	0.8863	3	0.7327	4	0.7943	4
7	1.1833	3	1.0124	3	0.7280	2	0.9833	4	0.7650	4	0.6788	3
8	1.2829	3	1.0595	3	0.8011	3	0.8868	4	0.7383	4	0.6874	3
9	1.2433	3	1.0501	3	0.8571	3	0.9303	4	0.6823	3	0.6575	3
10	1.2922	4	0.9971	2	0.8751	3	0.9115	4	0.6654	2	0.6269	3
11	1.0393	3	1.1844	4	0.7895	3	0.8720	3	0.7289	4	0.6975	3
12	1.0551	3	1.1529	4	0.8025	3	0.9771	4	0.7188	3	0.6596	3
13	1.1059	3	1.0482	3	0.8560	4	0.8625	4	0.7877	4	0.7537	4
14	1.2973	4	1.1419	4	0.8272	3	0.8771	3	0.8123	4	0.6260	3
15	1.2534	4	1.1095	3	0.8021	3	0.9849	4	0.8853	4	0.6971	3
16	1.1245	3	1.1616	4	0.8143	3	0.9947	4	0.7322	4	0.7277	4
17	1.2078	3	0.9770	3	0.7337	3	0.9437	4	0.7881	4	0.6536	3
18	1.0307	4	1.0180	2	0.8166	3	0.8598	3	0.8615	4	0.6998	3
19	1.2267	3	1.0835	2	0.7956	3	0.9644	4	0.8718	4	0.6473	3
20	1.2367	4	1.1481	4	0.8185	3	0.9735	4	<b>0.6418</b>	<b>2</b>	0.7238	4

\*Davis Boulder Index (DBI), \*\*Number of Clusters

**Table 7**  
The details of the experimental results for 20 runs of the IDBSCAN (Algorithm 2)

	Daily Demand Forecasting Orders		Istanbul Stock Exchange		Dow Jones Index		ADL Recognition		Software Engineering Teamwork		User Identification From Walking Activity	
	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**
1	0.9499	3	0.8919	2	0.7152	3	0.9596	3	0.6076	3	0.7239	4
2	0.7897	3	<b>0.7970</b>	<b>2</b>	0.7341	3	0.9136	3	0.6253	3	0.7159	4
3	0.9975	3	0.8528	3	0.6781	2	0.9219	3	0.7481	4	0.6311	3
4	0.882	2	0.9419	3	<b>0.5120</b>	<b>2</b>	<b>0.8010</b>	<b>2</b>	0.7453	4	0.7250	4
5	0.939	3	0.8240	3	0.5208	2	0.9352	3	0.6612	3	0.6356	3
6	0.8475	2	0.8667	2	0.6642	3	0.9466	3	0.6771	3	0.6095	2
7	0.7816	2	0.9029	4	0.5891	2	0.8730	2	0.7024	3	0.7004	3
8	0.9584	3	0.8641	3	0.5676	2	0.9792	4	<b>0.5723</b>	<b>2</b>	0.6202	3
9	0.8214	3	0.9380	4	0.5621	2	0.9944	4	0.7415	4	<b>0.5862</b>	<b>2</b>
10	<b>0.7777</b>	<b>2</b>	0.9714	4	0.5650	2	0.8032	2	0.5731	2	0.6473	3
11	0.7782	2	0.9824	4	0.5972	2	0.9095	3	0.7305	4	0.7033	3
12	0.7644	2	0.9737	4	0.6207	3	0.9421	3	0.6156	3	0.6557	2
13	0.8651	3	0.9534	3	0.6947	3	0.9859	4	0.5877	2	0.6351	2
14	0.9289	4	0.9487	3	0.6214	3	0.8078	2	0.7318	4	0.6972	3
15	0.9624	4	0.9400	3	0.6456	3	0.9831	4	0.7321	4	0.6411	3
16	0.8532	3	0.9185	3	0.7296	3	0.9160	3	0.6185	3	0.7415	4
17	0.8899	3	0.9349	3	0.5856	2	0.8289	2	0.7048	3	0.6050	2
18	0.9404	4	0.8578	2	0.7083	3	0.8625	2	0.6431	3	0.7358	3
19	0.8586	3	0.9067	3	0.5237	2	0.8503	2	0.6946	3	0.7372	3
20	0.9448	3	0.9851	3	0.6428	3	0.8104	2	0.5727	2	0.5885	2

\*Davis Boulder Index (DBI), \*\*Number of Cluster

**Table 8**

The details of the experimental results for 20 runs of the SIDBSCAN-Mamdani (Algorithm 3)

	Daily Demand Forecasting Orders		Istanbul Stock Exchange		Dow Jones Index		ADL Recognition		Software Engineering Teamwork		User Identification From Walking Activity	
	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**
1	0.8995	3	0.8222	3	0.5066	2	0.8325	4	0.5567	2	0.6325	4
2	0.8448	3	0.8765	3	0.5427	2	0.8880	4	0.6112	3	0.5418	3
3	0.8040	3	0.7826	2	0.5624	3	0.8290	4	0.6827	3	0.5394	3
4	0.7542	2	0.9496	4	0.6782	4	0.9114	5	0.7250	4	0.7082	5
5	0.7815	2	0.8165	3	0.6407	4	0.8643	4	0.6940	3	0.6238	4
6	0.8198	3	0.8866	3	0.5441	3	0.8417	4	0.6981	3	0.7167	5
7	0.8022	3	<b>0.7732</b>	<b>2</b>	0.5161	2	0.8375	5	0.7020	4	0.6651	4
8	<b>0.7279</b>	<b>2</b>	0.7858	2	0.6492	3	0.8753	5	0.5671	2	0.6639	4
9	0.8153	2	0.8645	4	0.6588	3	0.9587	5	0.7068	4	0.5820	3
10	0.8153	2	0.7905	2	0.5447	2	0.8721	5	0.5581	2	0.7093	5
11	0.8527	3	0.8845	4	0.6578	3	0.9902	5	0.6231	3	0.6038	4
12	0.8531	3	0.8959	4	0.5796	3	0.8719	4	0.6852	3	0.5439	3
13	0.8667	3	0.8369	3	0.6764	4	0.7838	3	0.6686	3	0.5324	3
14	0.8470	3	0.9615	4	0.6533	3	0.8121	3	0.7097	4	0.6017	4
15	0.8212	3	0.9328	4	0.5449	3	0.8971	3	0.5705	2	0.6414	4
16	0.7505	2	0.9170	4	0.5331	2	0.9877	4	0.5697	2	0.5267	3
17	0.8046	2	0.9818	4	0.5683	3	0.9106	5	<b>0.5549</b>	<b>2</b>	0.5832	4
18	0.8125	3	0.9657	4	0.6576	4	<b>0.7677</b>	<b>3</b>	0.5890	2	0.5924	4
19	0.8482	3	0.9745	4	0.5402	2	0.8989	4	0.5721	2	<b>0.5245</b>	<b>3</b>
20	0.8187	3	0.9277	4	<b>0.4890</b>	<b>2</b>	0.8926	4	0.5963	2	0.5903	4

\*Davis Boulder Index (DBI), \*\*Number of Clusters

**Table 9**

The details of the experimental results for 20 runs of the SIDBSCAN-Sugeno (Algorithm 3)

	Daily Demand Forecasting Orders		Istanbul Stock Exchange		Dow Jones Index		ADL Recognition		Software Engineering Teamwork		User Identification From Walking Activity	
	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**	DBI*	NC**
1	0.7393	2	0.8842	4	0.6156	3	0.7867	3	0.5733	2	<b>0.5201</b>	<b>3</b>
2	0.8141	3	0.7750	2	0.6806	4	0.7685	3	0.5900	3	0.6630	4
3	0.8824	4	0.8388	3	0.6152	3	0.8461	4	0.6837	4	0.5749	3
4	0.7680	2	0.8512	4	0.5044	2	<b>0.7654</b>	<b>3</b>	0.5888	2	0.6757	4
5	0.8131	3	0.8842	4	0.6641	3	0.8312	4	0.6072	3	0.6523	4
6	0.8622	3	0.8675	3	0.6948	4	0.8991	5	0.5977	2	0.5328	3
7	0.7720	3	0.8171	3	0.5382	2	0.8345	4	0.6150	3	0.6408	4
8	0.7187	2	0.8797	4	0.5214	2	0.8169	3	0.5708	2	0.5349	3
9	0.8531	3	0.8329	3	0.5966	3	0.8389	4	0.6670	4	0.5393	3
10	0.7684	2	0.7851	2	0.6360	3	0.7754	3	0.5828	2	0.6139	3
11	0.8867	4	0.8286	3	0.6112	3	0.8775	4	0.6691	3	0.6523	4
12	0.7171	2	0.7829	2	0.4670	2	0.7996	3	0.6782	3	0.5726	3
13	0.8028	3	0.8057	2	0.5313	3	0.7883	3	0.5528	2	0.6973	4
14	0.7413	2	0.8212	2	0.4894	2	0.8289	3	0.6738	4	0.6251	4
15	0.7686	2	0.7769	2	0.4948	2	0.7966	3	0.5940	2	0.6888	4
16	0.8150	3	0.8193	2	0.6360	3	0.8064	3	0.5683	2	0.5645	3
17	0.7781	2	0.8689	3	0.5709	2	0.8324	3	0.6140	3	0.6724	4
18	<b>0.7160</b>	<b>2</b>	0.8914	4	0.5883	3	0.8789	4	0.5757	2	0.6024	3
19	0.8123	3	<b>0.7676</b>	<b>2</b>	0.5297	3	0.8763	4	0.5602	2	0.5331	3
20	0.7502	2	0.8087	2	<b>0.4604</b>	<b>2</b>	0.7731	3	<b>0.5520</b>	<b>2</b>	0.5355	3

\*Davis Boulder Index (DBI), \*\*Number of Clusters

## References

1. Angelova, M., Pencheva, T. Tuning Genetic Algorithm Parameters to Improve Convergence Time. *International Journal of Chemical Engineering*, 2011. <https://doi.org/10.1155/2011/646917>
2. Aydilek, I. B., Arslan, A. A Hybrid Method for Imputation of Missing Values Using Optimized Fuzzy C-Means with Support Vector Regression and a Genetic Algorithm. *Information Sciences*, 2013, 25-35. <https://doi.org/10.1016/j.ins.2013.01.021>
3. Berkhin, P. A Survey of Clustering Data Mining Techniques. *Grouping multidimensional data*, 2006, 71.
4. Carvalho, D. R., Freitas, A. A. A Hybrid Decision Tree/Genetic Algorithm Method for Data Mining. *Information Sciences*, 2004, (1), 13-35. <https://doi.org/10.1016/j.ins.2003.03.013>
5. Ceci, M., A. Cuzzocrea, Malerba, D. Effectively and Efficiently Supporting Roll-up and Drill-Down Olap Operations over Continuous Dimensions via Hierarchical Clustering. *Journal of Intelligent Information Systems*, 2015, (3), 309-333. <https://doi.org/10.1007/s10844-013-0268-1>
6. Chaudhuri, S., Dayal, U. An Overview of Data Warehousing and Olap Technology. *ACM Sigmod Record*, 1997, (1), 65-74. <https://doi.org/10.1145/248603.248616>
7. Chen, J. Hybrid Clustering Algorithm Based on Pso with the Multidimensional Asynchronism and Stochastic Disturbance Method. *Journal of Theoretical and Applied Information Technology*, 2012, (1), 434-440.
8. Cheng, T. An Improved Dbscan Clustering Algorithm for Multi-Density Datasets. *Proceedings of the 2nd International Conference on Intelligent Information Processing*, 2017. <https://doi.org/10.1145/3144789.3144808>
9. Darong, H., Peng, W. Grid-Based Dbscan Algorithm with Referential Parameters. *Physics Procedia*, 2012, 1166-1170. <https://doi.org/10.1016/j.phpro.2012.02.174>
10. Davies, D. L., Bouldin, D.W. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, (2), 224-227. <https://doi.org/10.1109/TPAMI.1979.4766909>
11. Freitas, A. A., A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. *Advances in Evolutionary Computing*, 2003, 819-845. [https://doi.org/10.1007/978-3-642-18965-4\\_33](https://doi.org/10.1007/978-3-642-18965-4_33)
12. Gnanapriya, S., Suganya, R., Devi, G. S., Kumar, M. S. *Data Mining Concepts and Techniques*. *Data Mining and Knowledge Engineering*, 2010, (9), 256-263.
13. Han, J., Pei, J., Kamber, M. *Data Mining: Concepts and Techniques*. Elsevier, 2011.
14. Hema, R., Malik, N. *Data Mining and Business Intelligence*. *Proceedings of the 4th National Conference*, 2010.
15. Herrera, F., Lozano, M. *Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy, and Future Directions*. *Soft Computing*, 2003, (8), 545-562. <https://doi.org/10.1007/s00500-002-0238-y>
16. Herrera, F., Lozano, M. *Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers*. *Genetic Algorithms and Soft Computing*, 1996, 95-125.
17. Huang, Z. A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. *DMKD*, 1997, (8), 34-39.
18. Johnson, R. J., Williams, J. P., Bauer, K.W. *Autogad: An Improved Ica-Based Hyperspectral Anomaly Detection Algorithm*. *IEEE Transactions on Geoscience and Remote Sensing*, 2013, (6), 3492-3503. <https://doi.org/10.1109/TGRS.2012.2222418>
19. Joshi, A., Kaur, R. A Review: Comparative Study of Various Clustering Techniques in Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013, (3).
20. Karafotias, G., Hoogendoorn, M., Eiben, Á. E. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Transactions on Evolutionary Computation*, 2015, (2), 167-187. <https://doi.org/10.1109/TEVC.2014.2308294>
21. Karami, A., Johansson, R. Choosing Dbscan Parameters Automatically Using Differential Evolution. *International Journal of Computer Applications*, 2014, (7). <https://doi.org/10.5120/15890-5059>
22. Kumar, K. M., Reddy, A. R. M. A Fast Dbscan Clustering Algorithm by Accelerating Neighbor Searching Using Groups Method. *Pattern Recognition*, 2016, 39-48. <https://doi.org/10.1016/j.patcog.2016.03.008>
23. Liço, L. *Data Mining Techniques in Database Systems*.

24. Liu, J., Lampinen, J. A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Computing*, 2005, (6), 448-462. <https://doi.org/10.1007/s00500-004-0363-x>
25. Mamdani, E. H., Assilian, S. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 1975, (1), 1-13. [https://doi.org/10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
26. Mining, W. I. D. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.
27. Nagar, P., Srivastava, S. Application of Genetic Algorithms in Data Mining. 2nd National Conference on Challenges & Opportunities in Information Technology, 2008.
28. Pei, Z., Hua, X., Han, J. The Clustering Algorithm Based on Particle Swarm Optimization Algorithm. 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), 2008. <https://doi.org/10.1109/ICICTA.2008.421>
29. Pujari, A. K. *Data Mining Techniques*. Universities press, 2001.
30. Sautot, L., Faivre, B., Journaux, L., Molin, P. The Hierarchical Agglomerative Clustering with Gower Index: A Methodology for Automatic Design of Olap Cube in Ecological Data Processing Context. *Ecological Informatics*, 2015, 217-230. <https://doi.org/10.1016/j.ecoinf.2014.07.011>
31. Scovanner, P., Ali, S., Shah, M. A 3-Dimensional Sift Descriptor and Its Application to Action Recognition. *Proceedings of the 15th ACM International Conference on Multimedia*, 2007. <https://doi.org/10.1145/1291233.1291311>
32. Smiti, A., Eloudi, Z. Soft Dbscan: Improving Dbscan Clustering Method Using Fuzzy Set Theory. 2013 The 6th International Conference on Human System Interaction (HSI), 2013. <https://doi.org/10.1109/HSI.2013.6577851>
33. Smiti, A., Elouedi, Z. Dbscan-Gm: An Improved Clustering Method Based on Gaussian Means and Dbscan Techniques. 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), 2012. <https://doi.org/10.1109/INES.2012.6249802>
34. Takagi, T., Sugeno, M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. *Readings in Fuzzy Sets for Intelligent Systems*, 1993, 387-403. <https://doi.org/10.1016/B978-1-4832-1450-4.50045-6>
35. Vercellis, C. *Business Intelligence: Data Mining and Optimization for Decision Making*. John Wiley & Sons, 2011.
36. Woo, H. J., Joo, K. H., Park, N. H. A Clustering Olap Analysis in a Big Data Stream Environment, 2015. <https://doi.org/10.14257/astl.2015.99.23>
37. Zhang, H., Zhai, H., Zhang, L., Li, P. Spectral-Spatial Sparse Subspace Clustering for Hyperspectral Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 2016, (6), 3672-3684. <https://doi.org/10.1109/TGRS.2016.2524557>
38. Zhao, B., Zhu, Z., Mao, E., Song, Z. Image Segmentation Based on Ant Colony Optimization and K-Means Clustering. 2007 IEEE International Conference on Automation and Logistics, 2007. <https://doi.org/10.1109/ICAL.2007.4338607>
39. Zhao, Y.-Q., Yang, J. Hyperspectral Image Denoising Via Sparse Representation and Low-Rank Constraint. *IEEE Transactions on Geoscience and Remote Sensing*, 2015, (1), 296-308. <https://doi.org/10.1109/TGRS.2014.2321557>