Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 4, December 2014, pp. 256-267
http://dx.doi.org/10.5391/IJFIS.2014.14.4.256

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Automatic Switching of Clustering Methods based on Fuzzy Inference in Bibliographic Big Data Retrieval System

Maslina Zolkepli, Fangyan Dong, and Kaoru Hirota
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama, Japan

Abstract

An automatic switch among ensembles of clustering algorithms is proposed as a part of the bibliographic big data retrieval system by utilizing a fuzzy inference engine as a decision support tool to select the fastest performing clustering algorithm between fuzzy C-means (FCM) clustering, Newman-Girvan clustering, and the combination of both. It aims to realize the best clustering performance with the reduction of computational complexity from $O(n^3)$ to $O(n)$. The automatic switch is developed by using fuzzy logic controller written in Java and accepts 3 inputs from each clustering result, i.e., number of clusters, number of vertices, and time taken to complete the clustering process. The experimental results on PC (Intel Core i5-3210M at 2.50 GHz) demonstrates that the combination of both clustering algorithms is selected as the best performing algorithm in 20 out of 27 cases with the highest percentage of 83.99%, completed in 161 seconds. The self-adapted FCM is selected as the best performing algorithm in 4 cases and the Newman-Girvan is selected in 3 cases.The automatic switch is to be incorporated into the bibliographic big data retrieval system that focuses on visualization of fuzzy relationship using hybrid approach combining FCM and Newman-Girvan algorithm, and is planning to be released to the public through the Internet.

Keywords: Clustering, Fuzzy inference, Bibliographic big data, Visualization

1. Introduction

An automatic switch developed from fuzzy inference engine is a method that compares the result from 3 clustering methods based on their performance and selects the best performing algorithm to produce visualization result to the users. It is applied in the bibliographic big data retrieval system [1] that accepts user keywords to search for the relationship among bibliographic big data consisting of journal/conference papers.

Existing switching method includes a switch between classifiers [2] that suggests when to combine classifiers and how classifier selection effects the result. But the switch is not able to predict whether a combination of classifier can achieve a better result than individual classifier. A fuzzy inference system is used to assess the performance of a conventional power plant [3] that has a highly flexible algorithm as it can handle fuzzy data, crisp data, and data complexity. But it also uses singular clustering method as it performs better when compared to artificial neural network (ANN)-fuzzy C-mean (FCM) combination as it is able to explore performance patterns and select the better ones. Another fuzzy inference system that speeds up processes is

Received: Nov. 9, 2014
Revised : Dec. 1, 2014
Accepted: Dec. 8, 2014

Correspondence to: Corresponding author
(hirota@hrt.dis.titech.ac.jp)

global services of mobile communications churn management by using FCM and adaptive neuro fuzzy inference system [4] but the system only utilizes singular clustering method, and the complexity of the algorithm is high when the input volume is high. In the bibliographic big data retrieval system, to ensure the input volume can produce a result in less than 5 minutes [1], the dataset is generated upon each search command by the user to eliminate unrelated data, keeping the dataset small.

Previously, the bibliographic big data retrieval system only produces the visualization of the combination of both clustering algorithm, regardless whether it produces the best result or not. An automatic switch between ensembles of clustering methods is proposed as a part of a the system by utilizing a fuzzy inference engine as a decision support tool to select the fastest performing clustering algorithm between self-adapted FCM clustering [5], Newman-Girvan clustering algorithm [6], and the combination of both clustering methods [1]. The automatic switch accepts three inputs, which are the number of clusters, number of vertices in each cluster and the time required to complete the clustering process and produces the output in percentage for each clustering result. The clustering algorithms that has the smallest number of clusters and vertices and perform in the fastest time will get highest percentage, and is selected for visualization to the user.

Even though the best way to solve clustering problems is through a mixture of clustering algorithms, a hybrid clustering approach [2-4] requires higher computational complexity yet does not always produce the best result. Therefore the fuzzy inference engine targets to act as an automatic switch that compares the performance of each clustering algorithm where the clustering with the best performance will be selected for visualization to users. It aims to realize the best clustering performance with the reduction of computational complexity from $O(n^3)$ to $O(n)$, if the individual clustering algorithm is selected as the best performing algorithm.

The automatic switch is developed using jFuzzyLogic [7, 8], a fuzzy logic controller written in Java and the experiments are carried out in Eclipse IDE 4.2.2 [9] using Dell Latitude E5430 laptop with Intel Core i5-3210M at 2.50 GHz. The switch accepts 3 input variables for each clustering result and the values of the input variables from the clustering result are fuzzified according to the membership function, and then evaluated by 27 fuzzy inference rules. The evaluation result is defuzzified to get a crisp percentage output. The percentage output result of the fuzzy inference engine will determine which clustering method will be shown to the users in interactive visualization.

The dataset preparation from the result of three clustering methods are presented in Section 2. The application of fuzzy inference engine as an automatic switch between 3 clustering algorithms using jFuzzyLogic is presented in Section 3. Section 4 discusses the automatic switch experiment on clustering result and Section 5 describes the user feedback evaluation of the system.

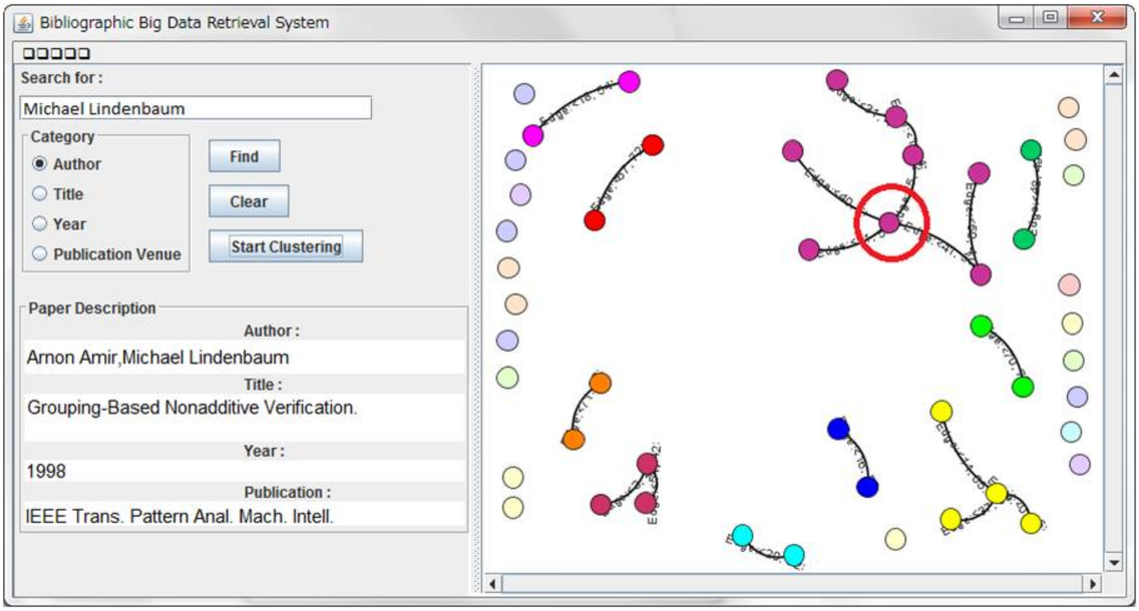
2. Clustering Result of the Self-Adapted FCM, Newman-Girvan Algorithm, and the Combination

The bibliographic big data retrieval system is developed to produce visualization of ensemble clustering result to the users. The systems user interface design is shown in Figure 1. A user will enter a keyword in the search box, select a category between paper author, title, year and publication venue, and click the ‘Find’ button. The system will search for related information in the MySQL database, and display the unclustered result on the right panel. Next, the user clicks the ‘Start Clustering’ button and the clustering process will be executed and by utilizing the automatic switch, the clustering with the best performance will be displayed to the user. To give users more information on each paper, when the user clicks on each vertex, the information of the paper will be displayed in the ‘Paper Description’ panel on the bottom left of the user interface.

In Figure 1, the keyword entered by the user is ‘Michael Lindenbaum’ for category ‘Author.’ The result consists of 86 papers that is written by the author. Out of 86 papers, there are 32 connections among the papers. When the node is clicked as shown in the red circle the information regarding the node is displayed in the ‘Paper Description’ box. Based on the visualization, it can be seen that the node has 4 connections with other nodes. It shows that this paper is one of the author’s important paper or key paper as it has many other papers that cites it.

To find relationship among bibliographic big data, the system uses a hybrid of clustering ensembles to improve clustering performance and to overcome the issue of weakness and strength of individual clustering algorithm. There are three types of clustering algorithms used in the system and results from the 3 clustering algorithms from the system is used as input for the automatic switch.

There is no absolute best criterion which would be independent of the final aim of the clustering [10]. To find the best performing clustering algorithm, several criteria have been determined that fulfills the objective of the target application. In



Search for :

Michael Lindenbaum

Category

● Author

○ Title

○ Year

○ Publication Venue

Paper Description

Author :

Arnon Amir,Michael Lindenbaum

Title :

Grouping-Based Nonadditive Verification.

Year :

1998

Publication :

IEEE Trans. Pattern Anal. Mach. Intell.

Figure 1. Bibliographic big data retrieval system's user interface.

the bibliographic big data retrieval system, to help users find the information they are looking for, the clustering result must be able to converge several important target papers that fulfills the users' search criteria. The time taken to produce the result should be less than 5 minutes [1]. Therefore, three criteria that can determine the desirable clustering performance are the time required to complete the clustering in seconds, number of related clusters found, and the total number of vertices found in the clusters.

Table 1 shows the result for all three clustering algorithms that contains the 3 information used as input variables for the automatic switch.

3. Automatic Switch between 3 Clustering Algorithms based on Fuzzy Inference

A fuzzy inference engine [11-13] is developed using jFuzzyLogic, a fuzzy logic controller written in Java [7, 8]. jFuzzyLogic is chosen to be the fuzzy language controller in the bibliographic big data retrieval system. jFuzzyLogic follows the standard for fuzzy control language and it provides an application programming interface and an Eclipse [9] plugin that simplifies the writing and testing of the FCL codes. The output generated by the fuzzy inference engine can easily be integrated into the bibliographic big data retrieval system as both are written in Java language.

Figure 2 describes the layout of the engine as an automatic switch. System inputs consists of three input variables, the number of related clusters (1-10 clusters), the total number of vertices in the related clusters (1-100 vertices), and the time required to get the result of the clustering process (1-300 seconds). The input variables are first fuzzified according to the input membership functions, then they will be evaluated by the fuzzy inference rules. Next, they will be defuzzify according to the output membership function that resulted in percentage (0%-100%) as the fuzzy inference output.

The system architecture of the bibliographic big data retrieval system is shown in Figure 3, where the automatic switch is situated between the clustering processes and the fuzzy visualization process. It shows that the automatic switch plays an important role to decide which clustering algorithms should be used in the interactive visualization to the user.

The fuzzification of the number of clusters is divided into 3 categories. Number of clusters is considered low if the amount is between 1-5 clusters, medium if it is between 1-10 clusters and high if 5-10 clusters are found. The fuzzification code for the fuzzy control language is described in Figure 4.

The membership degrees graph of the number of clusters is shown in Figure 5.

The fuzzification of the total number of vertices is also divided into 3 categories. The number of vertices found is considered few if the amount is between 1-25 vertices, several if

Table 1. Clustering result from bibliographic big data retrieval system

Keywords	N-G			SA-FCM			Combination		
	Time (s)	Clusters	Vertices	Time (s)	Clusters	Vertices	Time (s)	Clusters	Vertices
#1	110	6	24	204	2	9	248	1	5
#2	200	8	25	189	5	12	209	3	7
#3	152	4	8	165	4	12	178	3	7
#4	63	4	16	61	3	11	70	2	9
#5	265	10	58	259	8	36	271	6	32
#6	98	5	31	94	3	18	102	2	12
#7	173	7	26	171	5	18	176	4	14
#8	132	3	10	130	3	9	136	2	7
#9	159	3	11	156	2	9	161	1	4
#10	160	9	19	165	6	16	175	6	14
#11	69	7	22	61	7	19	76	7	13
#12	204	6	21	210	4	17	233	4	12
#13	167	4	10	189	3	8	204	3	6
#14	150	7	12	165	4	11	173	4	9
#15	160	5	16	165	3	14	170	3	11
#16	157	6	26	163	5	24	167	5	20
#17	194	6	33	195	5	31	210	5	25
#18	132	10	45	134	6	22	138	4	18
#19	133	4	15	137	2	8	141	2	6
#20	209	4	20	212	3	15	231	3	11
#21	65	7	21	67	4	17	71	4	11
#22	123	8	32	129	5	27	137	5	13
#23	135	8	22	151	7	21	159	7	20
#24	116	9	36	122	6	31	128	6	22
#25	102	4	21	113	4	19	119	3	14
#26	127	5	28	138	4	25	143	3	11
#27	159	4	11	160	3	10	169	2	7
Avg.	145	6	23	152	4	17	163	4	13

FCM, fuzzy C-mean.

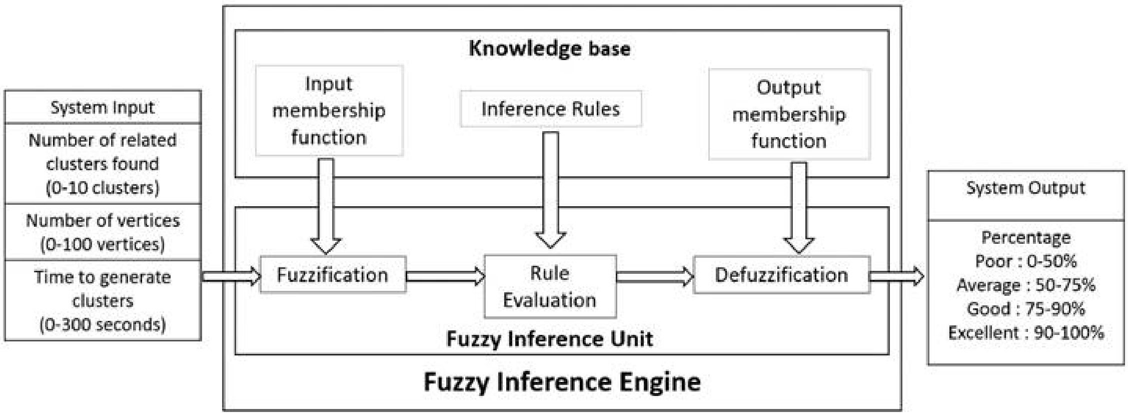


Figure 2. The automatic switch layout.

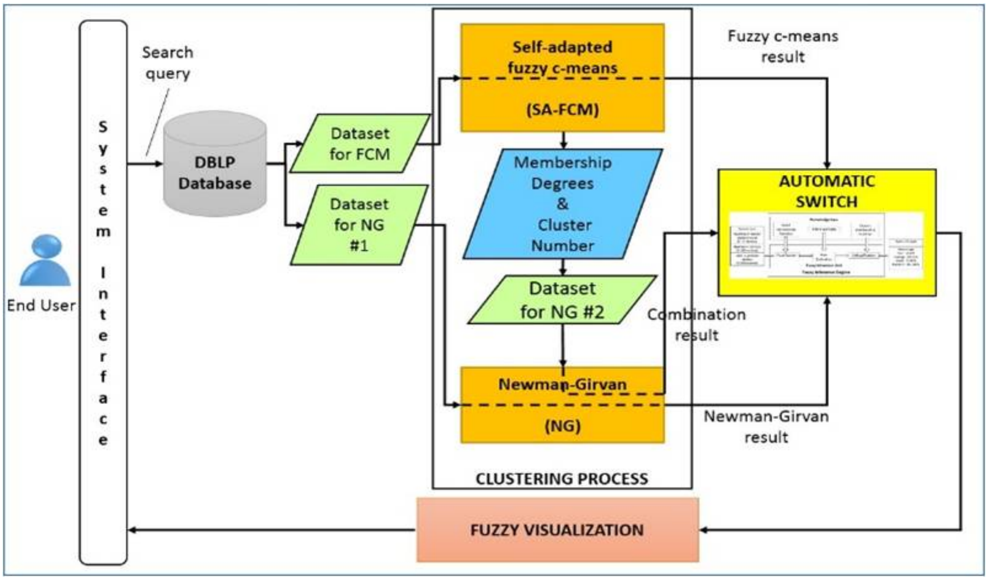
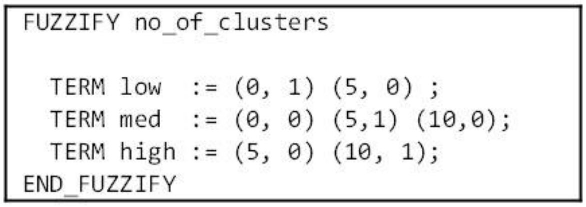


Figure 3. Bibliographic big data retrieval system architecture with automatic switch function.



```
FUZZIFY no_of_clusters

  TERM low  := (0, 1) (5, 0) ;
  TERM med  := (0, 0) (5,1) (10,0);
  TERM high := (5, 0) (10, 1);
END_FUZZIFY
```

Figure 4. Fuzzification of total number of clusters.

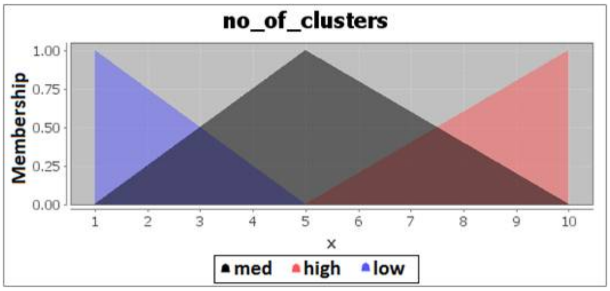
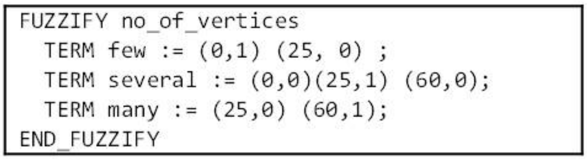


Figure 5. Membership degrees graph for total number of clusters.

the amount is between 1-60 vertices, and many if the amount is between 25-60 vertices. The fuzzification code for number of vertices is shown in Figure 6 and the membership degrees graph for vertices is shown is Figure 7.

For time required to complete the clustering process, there are also 3 categories to represent them. Time is considered short if it takes between 1 to 200 seconds to complete the process,



```
FUZZIFY no_of_vertices
  TERM few := (0,1) (25, 0) ;
  TERM several := (0,0)(25,1) (60,0);
  TERM many := (25,0) (60,1);
END_FUZZIFY
```

Figure 6. Fuzzification of total number of cluster.

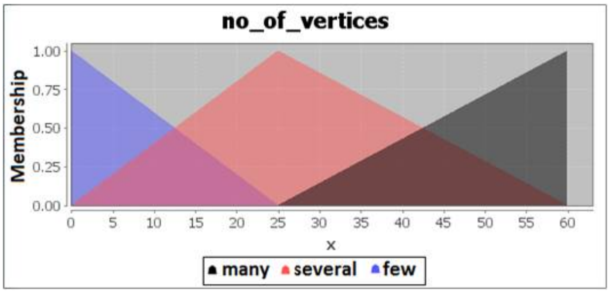
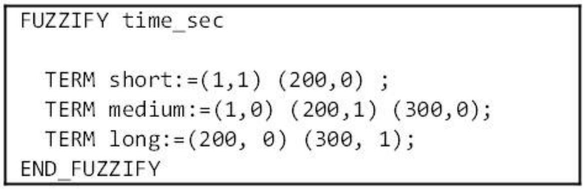


Figure 7. Membership degrees graph for total number of vertices.

medium if it is between 1-300 seconds, and it is considered long if it takes between 200-300 seconds to complete the clustering process. The fuzzification code for time is shown in Figure 8 and the membership graph is shown in Figure 9.

The defuzzification process are performed to get a non-fuzzy value that best represents the possibility distribution of an inferred fuzzy control action. The defuzzified output will be in percentage, where the clustering result with the highest percent-



```
FUZZIFY time_sec

  TERM short:=(1,1) (200,0) ;
  TERM medium:=(1,0) (200,1) (300,0);
  TERM long:=(200, 0) (300, 1);
END_FUZZIFY
```

Figure 8. Fuzzification of time in seconds.

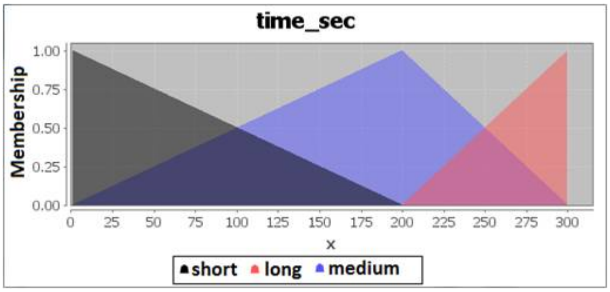


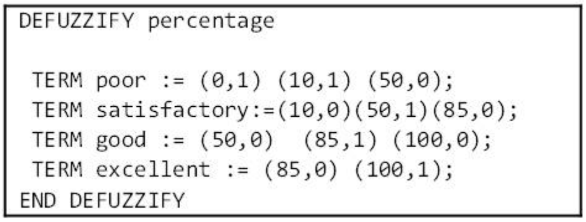
Figure 9. Membership degrees graph for time in seconds.

age will be selected for visualization to the end user. There are four categories for the percentage output, ranging from poor, average, good to excellent. Poor percentage ranges from 0%-50%, average percentage ranges from 10%-85%, good percentage ranges from 50%-100% and excellent percentage ranges from 85%-100%. The defuzzification code for percentage control is shown in Figure 10 and the membership degree graph for percentage is shown in Figure 11.

The defuzzification strategy used in the proposed method is the center of gravity or centroid method. The strategy is the most common and physically appealing of all the defuzzification methods[14, 15] It is given by the algebraic expression where $\int$ denotes an algebraic integration, as shown in

$$\int x\mu(x)dx / \int \mu(x)dx. \tag{1}$$

There is no systematic procedure for choosing a good defuzzification strategy and it depends on the properties of the application. Therefore, the center of gravity strategy is chosen due to its computational simplicity where it does not require complex computation that may lead to more time. Since the end result which is the visualization of bibliographic data search result needs to be produced in less than 5 minutes, this is an important criteria to keep the calculation process time as short as possible.



```
DEFUZZIFY percentage

 TERM poor := (0,1) (10,1) (50,0);
 TERM satisfactory:=(10,0)(50,1)(85,0);
 TERM good := (50,0)  (85,1) (100,0);
 TERM excellent := (85,0) (100,1);
END DEFUZZIFY
```

Figure 10. Defuzzification for percentage output and defuzzification method specification.

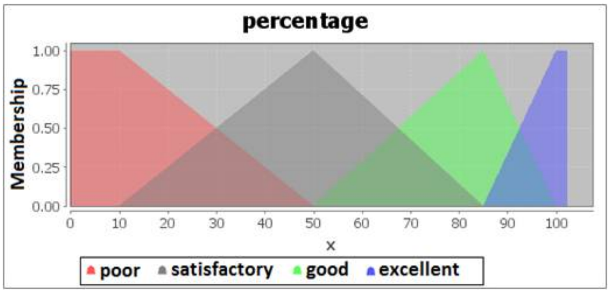
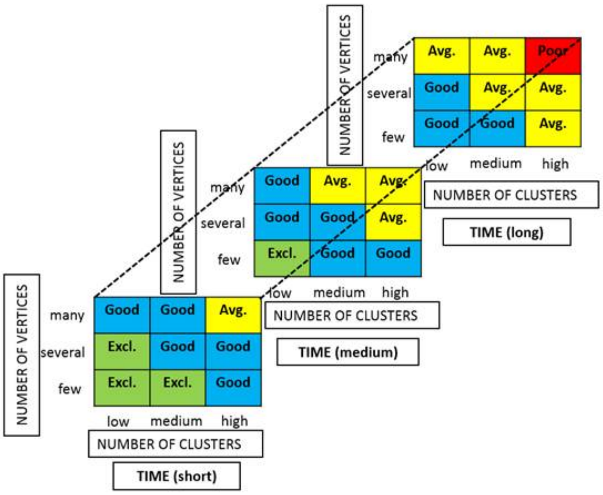


Figure 11. Membership degrees graph for percentage.



TIME (short)

NUMBER OF VERTICES \ NUMBER OF CLUSTERS	low	medium	high
many	Good	Good	Avg.
several	Excl.	Good	Good
few	Excl.	Excl.	Good

TIME (medium)

NUMBER OF VERTICES \ NUMBER OF CLUSTERS	low	medium	high
many	Good	Avg.	Avg.
several	Good	Good	Avg.
few	Excl.	Good	Good

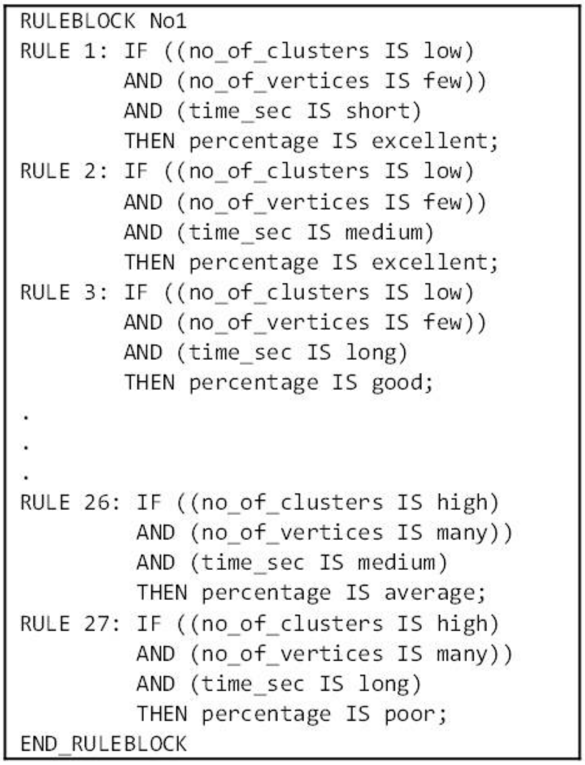
TIME (long)

NUMBER OF VERTICES \ NUMBER OF CLUSTERS	low	medium	high
many	Avg.	Avg.	Poor
several	Good	Avg.	Avg.
few	Good	Good	Avg.

Figure 12. Fuzzy associative matrix for percentage control.

A fuzzy associative matrix is developed to cover every possible outcome from the clustering result. Figure 12 shows the fuzzy associative matrix for the clustering results with three inputs, number of clusters, number of vertices and time. Inside each box is written a label of the automatic switch output. In the automatic switch, there are 27 possible rules corresponding to the 27 boxes in the matrix.

Figure 13 describes the rule block that consists 27 inference



```
RULEBLOCK No1
RULE 1: IF ((no_of_clusters IS low)
        AND (no_of_vertices IS few))
        AND (time_sec IS short)
        THEN percentage IS excellent;
RULE 2: IF ((no_of_clusters IS low)
        AND (no_of_vertices IS few))
        AND (time_sec IS medium)
        THEN percentage IS excellent;
RULE 3: IF ((no_of_clusters IS low)
        AND (no_of_vertices IS few))
        AND (time_sec IS long)
        THEN percentage IS good;
.
.
.
RULE 26: IF ((no_of_clusters IS high)
         AND (no_of_vertices IS many))
         AND (time_sec IS medium)
         THEN percentage IS average;
RULE 27: IF ((no_of_clusters IS high)
         AND (no_of_vertices IS many))
         AND (time_sec IS long)
         THEN percentage IS poor;
END_RULEBLOCK
```

Figure 13. Rule block for inference rules of the automatic switch.

rules for the automatic switch in fuzzy language control codes.

4. Experiment of Automatic Switching on Clustering Results

The percentage result from the automatic switch is shown in Table 2. Out of the three clustering result, the combination of both clustering algorithms has the highest mean percentage of 76.06%, as opposed to 76.02% for self-adapted FCM algorithm and 70.72% for Newman-Girvan algorithm. It shows that on average, the combination always perform better than the individual clustering algorithms. This can be seen from the average number of clusters and vertices produced by each clustering algorithm as shown on Table 1, where the combination always generates the least number of clusters and vertices as opposed to the individual clustering algorithms, in under 5 minutes.

The self-adapted FCM has the smallest standard deviation of 6.31% which means that its performance does not vary greatly as opposed to the Newman-Girvan and the combination algorithm. This is because the self-adapted FCM is more flexible than the crisp Newman-Girvan algorithm due to its fuzzy properties. Newman-Girvan algorithm has the highest standard deviation of 11.151%. The average percentage for Newman-Girvan is 70.72%, but it performs poorly for keyword #5 with 33.51%. For this keyword, Newman-Girvan takes 265 seconds to find 58 vertices and 10 clusters. Based on the fuzzy inference rules of the automatic switch, a low percentage is given to the Newman-Girvan algorithm as it does not favor the users of the system who requires more time to examine each of the 58 vertices for their desired papers. The combination sits in the middle with a standard deviation of 6.88%. The performance of the combination varies greater than the self-adapted FCM because it combines both clustering algorithms therefore also inherits their less than efficient performance.

Table 2. Automatic switch result – percentage output

Keyword	Percentage (%)		
	Newman-Girvan	Self-adapted fuzzy C-means	Combination
#1	77.56	76.04	68.99
#2	62.39	77.94	72.07
#3	78.74	78.65	79.48
#4	78.74	79.42	80.16
#5	33.51	50.42	53.54
#6	66.41	79.21	79.09
#7	76.11	78.41	78.72
#8	79.48	79.48	80.93
#9	79.40	80.10	83.99
#10	54.70	77.91	77.59
#11	77.84	78.24	78.94
#12	75.40	70.89	62.26
#13	78.74	79.48	75.75
#14	77.79	78.72	78.74
#15	78.26	78.78	79.40
#16	76.04	78.14	78.17
#17	65.99	67.72	71.12
#18	54.88	77.93	78.18
#19	78.60	80.58	81.05
#20	70.98	70.22	62.64
#21	77.98	78.74	79.40
#22	61.88	73.92	78.07
#23	62.50	78.18	78.35
#24	56.15	67.01	77.84
#25	78.63	78.73	78.78
#26	72.05	78.18	79.40
#27	78.72	79.48	81.09
Mean	70.72	76.02	76.06
Std. Dev.	11.15	6.31	6.88

Based on the bar chart for percentage shown in Figure 14, the combination algorithm performance shown in grey line

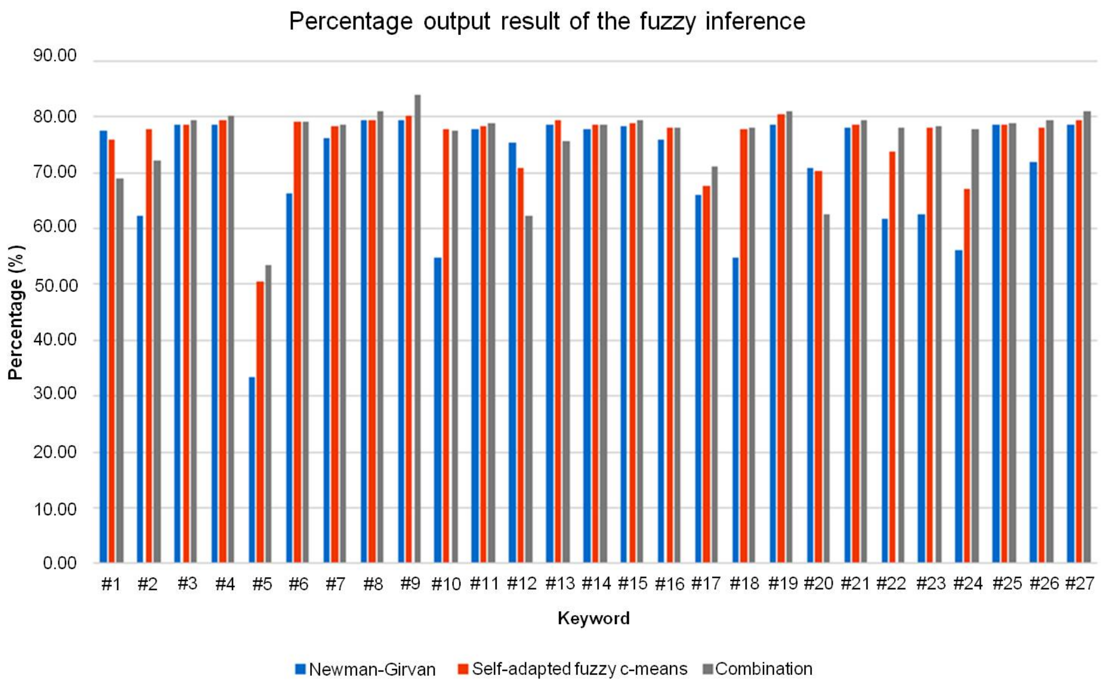


Figure 14. Bar chart for percentage output result of the fuzzy inference.

performs better in most search cases. It received the highest percentage in 20 out of 27 search cases with the highest percentage of 83.99% for keyword #9. For keyword #9, the combination algorithm successfully gathered 4 target papers in one cluster in 161 seconds. The few number of papers, and the short time to produce the result gives the combination algorithm the highest percentage as compared to the other 2 clustering algorithms. The combination algorithm do not get the highest percentage in 7 cases (keywords #1, #2, #6, #10, #12, #13, and #20) as the keywords return a larger search result compared to the other keywords thus more time are required to perform the clustering process on them. According to the fuzzy inference rules, if the time taken to produce the clustering result is long (200-300 seconds), the percentage is will not be in 'Excellent' category. Therefore, the application of the automatic switch will be able to compare the performance and select the best performing algorithm each time a search is performed.

The Newman-Girvan algorithm's computable complexity is O(n) and the self-adapted FCM algorithm has a computable complexity of $O(n^3)$. When combined, the complexity of the algorithms becomes $O(n^3)$, at worst. Therefore, by applying the automatic switch to compare the performance of all three algorithms, if the Newman-Girvan algorithm is selected as the best performing algorithm, the computable complexity to produce the visualization of the clustering result is reduced from $O(n^3)$ to $O(n)$.

5. User-Based Evaluation for the Bibliographic Big Data Retrieval System

To evaluate the system's usability, user-based evaluation method is used where the system is tested by selecting 18 participants to perform a set of pre-determined tasks on the system prototype. A feedback questionnaire is given to the participants after the tasks have been performed on the system prototype and the participants have to fill in the questionnaire based on their opinion of the system. This method is used as it is the most realistic estimate of usability [16].

A 16-question feedback questionnaires focusing on bibliographic visualization tool objectives is designed to evaluate the usability of the system. There are 4 categories covered in the questionnaire, which are the content organization, navigation, graphical user interface, and effectiveness and performance of the system.

Questions for evaluating the system's usability are:

1) Content organization
   - Displays complete bibliographic entry

- Displays chronology of paper on request
- Displays influence of article on other articles
- Displays publication information by fields of knowledge
- Displays strength of relationship between articles
- Shows relationship between research areas

2) Navigation

- Provides exploration of activities of a particular author
- Filters information by user's request
- Offers comfortable navigation methods
- Provides wide range of options to explore different part of bibliographic data

3) Effectiveness and performance

- Effectively express relationships contained in bibliographic data
- Search result is visualized in 5 minutes or less
- Provides easy understanding of relationship between researchers
- Provides user with good control over the information to be displayed

4) Graphical user interface

- Good graphical design
- Attractive presentation

Five options are given to the participants for each question. The participants will choose one option from 1=Highly Dissatisfied, 2=Dissatisfied, 3=Neutral, 4=Satisfied, to 5=Highly Satisfied. The feedback result is analyzed using WEBUSE [17], a website usability evaluation tool. The answer options and their corresponding merits are shown in Table 3.

Usability point for a category, x, is defined in

$$x = \left(\sum m\right)/q, \tag{2}$$

where m is the merit for each question and q is the number of question for each category.

Table 4 shows the usability levels and the corresponding usability points.

A total number of 18 participants have volunteered to test the prototype of the bibliographic big data retrieval system where 44.4% are males and 55.6% are female participants. 22.2% of the participants are undergraduate students and 77.8% are postgraduate students with 38.9% have between 6 and 10 years of computer experience, and 61.1% have more than 10 years of computer experience. Only 16.7% of the participants have

Table 3. Answer option for feedback questionnaire and corresponding merits

Option	Merit
Highly dissatisfied	1.00
Dissatisfied	0.75
Neutral	0.50
Satisfied	0.25
Highly satisfied	0.00

Table 4. Usability points and corresponding usability levels

Points, x	Usability level
$0 \leq x \leq 0.2$	Bad
$0.2 < x \leq 0.4$	Poor
$0.4 < x \leq 0.6$	Moderate
$0.6 < x \leq 0.8$	Good
$0.8 < x \leq 1.0$	Excellent

Table 5. System evaluation result by participants

	Scale level (%)				
Usability category	Highly dissatisfied	Dissatisfied	Neutral	Satisfied	Highly satisfied
Content organization	3.7	6.4	21.3	48.1	20.4
Navigation	2.8	8.3	5.5	47.2	36.1
User interface	5.5	11.1	19.4	37.5	26.4
Performance & effectiveness	0	2.8	30.5	36.1	30.5
Total	3.4	7.6	18.1	43.8	27.1

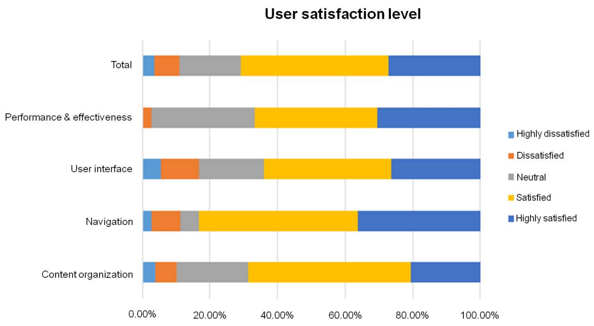


Figure 15. Bar chart for user satisfaction level.

Table 6. Usability level and usability points for each category

Usability category	Usability point	Usability level
Content organization	0.6875	Good
Navigation	0.7639	Good
User interface design	0.6701	Good
Performance & effectiveness	0.7361	Good
Total	0.6875	Good

used another bibliographic visualization tool before taking part in this survey.

Table 5 shows a summary of user satisfaction level in 4 categories of usability from content organization, navigation, user interface design, and performance and effectiveness. From the satisfaction scale level, users are most satisfied in the navigation aspect of the system with 36.1% of participants highly satisfied with the navigation styles that the system offers. The bar chart for the user satisfaction level is shown in Figure 15. It shows that the user satisfaction level mainly falls in the 'Satisfied' category as indicated in yellow.

Table 6 shows a summary of usability point and usability level based on each category. From the overall usability level, it can be concluded that the system is accepted as "Good" based on the usability scale. System's navigation and performance receives a high point of 0.7639 and 0.7361 respectively. It shows that the users are satisfied with the system's navigation and performance but improvements are necessary to increase the usability level from 'Good' to 'Excellent.' The system's content organization and user interface design receives usability points less than 0.7 therefore even though the user is somewhat satisfied with these aspects of the system, more improvements are needed to be done to increase users' satisfactions on the usability of the system as a whole.

Some comments given by participants to improve the content organization is that the system should not only give the best clustering performance but also shows the number of citation of evey authors or papers selected by users. It is an important information that gives extra weight to every author and paper nodes that they wish to explore.

Other comments state that the system should offer the users an option to control the importance of their search, whether they want only highly related information to be shown or to include a bigger search result that shows all related results, whether they are highly related or somewhat related to the input information.

Obtaining multiple levels of useful information from large amounts of data requires scalable algorithms to produce timely results [18]. Current algorithms are inefficient in terms of big data analysis. the bigger the data gets the less efficient each algorithm will perform, and leads to higher computational complexity.

To increase the performance of clustering algorithms to process big data, efficient tools and technologies are essential to process such data. Currently the dataset used in the bibliographic big data retrieval system is stored in MySQL 5.6, an open source relational database management system. The issue with current database while it is suitable for research purposes, to ensure faster and more efficient service, a database specifically designed for big data is preferable.

6. Conclusions

The three criteria that determine the desirable clustering performance in the bibliographic big data retrieval system are the time required to complete the clustering in seconds, number of related clusters found, and the total number of vertices found in the clusters. The automatic switch accepts these three criteria as its input and the experiment is carried out in Eclipse IDE 4.2.2, connected to MySQL 5.6 database that stores the bibliographic big data, using Dell Latitude E5430 laptop with Intel (R) Core (TM) i5-3210M at 2.50 GHz. The experimental result demonstrates that the combination of both clustering algorithms is selected as the best performing algorithm in 20 out of 27 cases with the highest percentage of 83.99%, completed the process in 161 seconds. The self-adapted FCM is selected as the best performing clustering algorithm in 4 search cases with the highest percentage at 80.58%, completed in 137 seconds and Newman-Girvan algorithm is selected in 3 search cases with the highest percentage at 79.46% in 132 seconds. By applying the automatic switch in the bibliographic big data retrieval system, the best performing algorithm can be determined in every search case executed by the users. The computable complexity of the self-adapted FCM and the combination algorithm is O(n^3), while Newman-Girvan is O(n). For every search cases that Newman-Girvan is selected as the best performing algorithm, the computational complexity of the clustering process is reduced as it will only produce the visualization result of Newman-Girvan clustering result to the users. The self-adapted fuzzy c-means and the Newman-Girvan algorithm are selected to be combined because their features complement each other and they are suitable for bibliographic big data retrieval application[1]. The combination algorithms performance is compared to the two singular clustering algorithms using the automatic switch to ensure that the best performing algorithm is chosen for each search case. Other types of clustering methods can also be considered to be applied to the automatic switch, with little modification to fulfill the purpose of the application target. The feedback survey shows the overall level of system usability is good and acceptable to users. Users are satisfied with the navigation, effectiveness, and performance of the system but some improvements needs to be done to increase the system's usability especially in terms of content organization and user interface design.

The automatic switch is incorporated into the bibliographic big data retrieval system that focuses on visualization of fuzzy relationship using hybrid approach combining the salf-adapted FCM and Newman-Girvan algorithm. The system is currently being developed and improved. Future works includes:

1) emphasizing the visualization of fuzzy relationship to differentiate from crisp clustering relationship to effectively display the fuzzy visualization result to the users; and
2) planning to be released to the public through the Internet.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

[1] M. Zolkepli, F. Dong, and K. Hirota, "Visualization of fuzzy relationship using clustering algorithms in bibliographic big data," in *Proceedings of the 14th International Symposium on Advanced Intelligent Systems*, Daejeon, Korea, November 13-16, 2013.

[2] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 2, pp. 146-156, Apr. 2002. http://dx.doi.org/10.1109/3477.990871

[3] A. Azadeh, V. Ebrahimipour, and P. Bavar, "A fuzzy inference system for pump failure diagnosis to improve maintenance process: the case of a petrochemical industry," *Expert Systems with Applications*, vol. 37, no. 1, pp. 627-639, Jan. 2010. http://dx.doi.org/10.1016/j.eswa.2009.06.018

[4] A. Karahoca and D. Karahoca, "GSM churn management by using fuzzy C-means clustering and adaptive neuro fuzzy inference system," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1814-1822, Mar. 2011. http://dx.doi.org/10.1016/j.eswa.2010.07.110

[5] J. Jin, Y. Liu, L. T. Yang, N. Xiong, and F. Hu, "An efficient detecting communities algorithm with self-adapted fuzzy C-means clustering in complex networks," in *Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, UK, June 25-27, 2012, pp. 1988-1993. http://dx.doi.org/10.1109/TrustCom.2012.76

[6] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821-7826, Jun. 2002. http://dx.doi.org/10.1073/pnas.122653799

[7] P. Cingolani and J. Alcal-Fdez, "jFuzzyLogic: a Java library to design fuzzy logic controllers according to the standard for fuzzy control programming," *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 61-75, Jun. 2013. http://dx.doi.org/10.1080/18756891.2013.818190

[8] P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: a robust and flexible fuzzy-logic inference system language implementation," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Brisbane, QLD, June 10-15, 2012, pp. 1-8. http://dx.doi.org/10.1109/FUZZ-IEEE.2012.6251215

[9] The Eclipse Foundation, "Eclipse IDK 4.2.2," Available http://www.eclipse.org

[10] "A tutorial on clustering algorithms," Available http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

[11] V. Cherkassky, "Fuzzy inference systems: a critical review," in *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications (NATO ASI Series Volume 162)*, O. Kaynak, L. Zadeh, B. Trken, and I. Rudas, Eds. Heidelberg: Springer Berlin, 1998, pp. 177-197. http://dx.doi.org/10.1007/978-3-642-58930-0_10

[12] O. Wolkenhauer, "Fuzzy inference engines," in *Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis*, O. Wolkenhauer, Ed. New York, NY: John Wiley & Sons, 2002, pp. 161-172. http://dx.doi.org/10.1002/0471224340.ch8

[13] R. Rojas, "Fuzzy logic," in *Neural Networks*, Heidelberg: Springer Berlin, 1996, pp. 287-308. http://dx.doi.org/10.1007/978-3-642-61068-4_11

[14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 1, pp. 116-132, Jan. 1985. http://dx.doi.org/10.1109/TSMC.1985.6313399

[15] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. II," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 419-435, Mar. 1990. http://dx.doi.org/10.1109/21.52552

[16] A. Dillon, "Usability evaluation," in *International Encyclopedia of Ergonomics and Human Factors*, W. Karwowski, Ed. New York, NY: Taylor & Francis, 2001.

[17] T. K. Chiew and S. S. Salim, "Webuse: website usability evaluation tool," *Malaysian Journal of Computer Science*, vol. 16, pp. 47-57, Jun. 2003.

[18] D. Talia, "Clouds for scalable big data analytics," *Computer*, vol. 46, no. 5, pp. 98-101, May 2013. http://dx.doi.org/10.1109/MC.2013.162

Maslina Zolkepli received the M.S. degree of computer science from Universiti Putra Malaysia in 2010. Currently she is a doctoral candidate at the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology (Tokyo Tech), Japan. Her research interests include computational intelligence, data mining, and fuzzy logic. She is a member of Japan Society for Fuzzy Theory and Intelligent Informatics since 2012.

Fangyan Dong received Dr. E. degree from Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology (Tokyo Tech), Japan, in 2003. Since 2003, she has been with Tokyo Institute of Technology as a post-fellow researcher, an assistant professor, and currently is an associate professor of both Education Academy of Computational Life Sciences (ACLS) and Department of Computational Intelligence and Systems Science at Tokyo Institute of Technology. Her research interests include computational intelligence, logistics optimization, Kansei engineering, and intelligent robot. She is members of Japan Society for Fuzzy Theory and Intelligent Informatics, Japanese Society for Artificial Intelligence, and Information Processing Society of Japan. She published 75 journal papers and 130 conference papers, and received 10 awards.

Kaoru Hirota received Dr. E. degree from Tokyo Institute of Technology in 1979. After his career at Sagami Institute of Technology and Hosei University, he has been with Tokyo Institute of Technology. His research interests include fuzzy systems, intelligent robot, and image understanding. He experienced president and fellow of International Fuzzy Systems Association (IFSA), and president of Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT). He is a chief editor of Journal of Advanced Computational Intelligence and Intelligent Informatics. Banki Donat Medal, Henri Coanda Medal, Grigore MOISIL Award, SOFT best paper award, Acoustical Society of Japan best paper award, honorary/adjunct professorships from "de La Salle University (Philippine), Changchun University of Science & Technology (China), Harbin University of Science and Technology (China), the University of Nottingham (UK), and Beijing Institute of Technology (China)", and Honoris Causa from "Bulacan state university (Philippine), Budapest Technical University (Hungary), and Szechenyi Istvan University (Hungary)" were awarded to him. He organized more than 10 international conferences/symposiums as a founding/ general/program chair. He has been publishing more than 250 journal papers, 50 books, and 500 conference papers.