Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Designing Enhanced Automatic Genetic Clustering Algorithm for Unknown Number of Clusters: An Experimental Evaluation

Muhamad Hariz Muhamad Adnan[1,2,*], Mohd Fadzil Hassan[3], Rishi Kumar[3], Nurul Akhmal Mohd Zulkefli[4], Chee Ken Nee[1,2]

[1] Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia
[2] Center of Digital Transformation in Education Technopreneurship, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia
[3] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perakk, Malaysia
[4] College of Arts and Applied Sciences, Dhofar University, Salalah, Oman

ARTICLE INFO

Article history:
Received 9 January 2025
Received in revised form 14 February 2025
Accepted 4 April 2025
Available online 30 April 2025

Keywords:
Genetic algorithm; clustering; cluster analysis; optimization; cloud service

ABSTRACT

This research seeks to develop an improved automatic genetic clustering technique to support the negotiation of heterogeneous and multi-attribute cloud services. It is known that heterogeneous cloud services involve cloud service providers offering varying sizes and costs for cloud service virtual machines, processors and storage. Existing automatic clustering techniques, such as the Automatic Genetic Clustering Algorithm for Unknown Number of Clusters (GCUK) algorithm, can provide the solution for heterogeneous cloud services negotiation, but they can also result in sub-optimal clusters, overlapping and partial solutions, local optima trapping and sub-optimal chromosome size. The automatic clustering technique employs the modified GCUK algorithm to address this issue. It was intended to solve GCUK constraints, local optima, imperfect clustering and suboptimal chromosomal size, as well as to support different cloud services. The findings indicate that the improved GCUK algorithm can provide optimal clustering solutions and avoid local optima, achieve optimal accuracy and resilience and obtain dynamic and optimal chromosomal sizes at runtime.

1. Introduction

One of the main goals of data clustering is to divide a set of data into groups (clusters) based on their natural similarities. A cluster is a group of objects that are "homogeneous" among themselves but "heterogeneous" concerning objects in other clusters [5]. Clustering is a great way to work with large amounts of data and look for patterns in the data set. Clustering tries to make sure that the data within a cluster are the most alike and that the data from different clusters are the least alike [11]. The scientific community agrees that K-means is the most popular clustering algorithm. This is because its results are easy to understand and there are different ways to use it [4].

Clustering is a type of unsupervised learning, in which the learning algorithm is not given any labels and is left to find patterns on its own in the data it is given. It is important for unlabelled data

* *Corresponding author*
E-mail address: mhariz@meta.upsi.edu.my

https://doi.org/10.37934/ard.129.1.101111

where there is no target variable to predict. Even though there are many classical clustering algorithms, most of them have major flaws, such as being sensitive to initial cluster centres, which makes it easy for them to get stuck in local optimum solutions. The other big problem with data clustering algorithms is that they can't be made to work the same way for everyone. On the other hand, clustering can be thought of as an optimization, which is an NP-hard optimization that is hard to solve [5].

In clustering algorithms that are currently used, the number of clusters must be given ahead of time. And, for a given number of clusters K, K has a big effect on how clustering works. There is no one good way to figure out what the value of cluster number K is [19]. Most well-known clustering methods based on distance measures, distance metrics and similarity functions have the problem of getting stuck in local optima and their performance strongly depends on the initial values of the cluster centres [9].

In this research paper, the clustering techniques were investigated to solve the heterogeneous cloud service negotiation optimization problem [14-16]. These works revealed that some of the double auction frameworks and mechanisms utilized the Continuous Double Auction (CDA) to negotiate one or two cloud service attributes and showed high execution time. Some mechanisms were reported to have a long execution time due to the utilization of certain algorithms.

Going back to basics, the Automatic Genetic Clustering Algorithm for unknown number of clusters algorithm (CGUK) was investigated [7]. The GCUK has been tested with a complete evaluation using synthetic and real data sets as well as the application field [12]. The GCUK is simple in using the standard GA operators such as population initialization, fixed chromosome size encoding, crossover and mutation. Hence it was selected for clustering [10].

The further study identified that the GCUK produced overlapping and partial clustering solutions on heterogeneous cloud service datasets from the preliminary experiments. Some of the GCUK chromosomes have sub-optimal clusters where the solutions are trapped in local optima [10]. The GCUK has been referred to by many new algorithms [6,8,13,17,18,20]. Enhancing the GCUK to address the limitations of cloud service negotiation is important research. Therefore, this paper aims to design an enhanced GCUK algorithm and validate the performance of the proposed algorithm.

2. Methodology

To design and propose the improved GCUK algorithm, this study technique proposed five tasks. Figure 1 depicts the depicted activities. The specified five activities are literature review (investigation of GCUK), cloud service marketplace investigation, data formulation, automatic clustering algorithm development and validation.

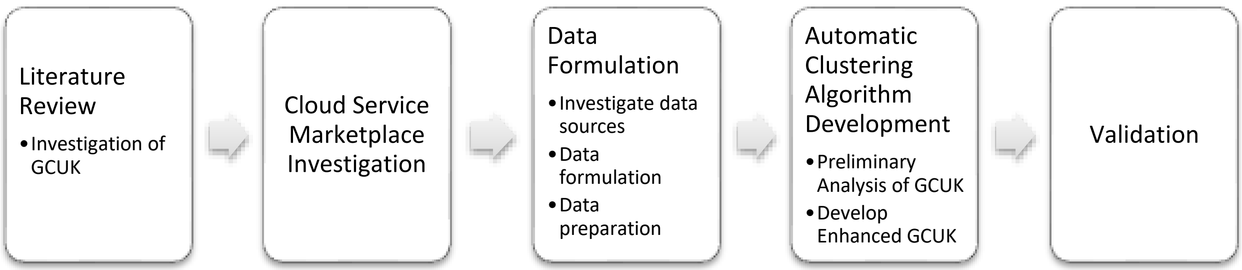

Fig. 1. The research activities

The GCUK algorithm is enhanced in the population initialization and genetic operations stages. The proposed enhanced GCUK algorithm is shown in the flowchart in Figure 2. The pseudocodes of the original GCUK algorithm and the proposed enhanced GCUK algorithm are compared in Figure 3.

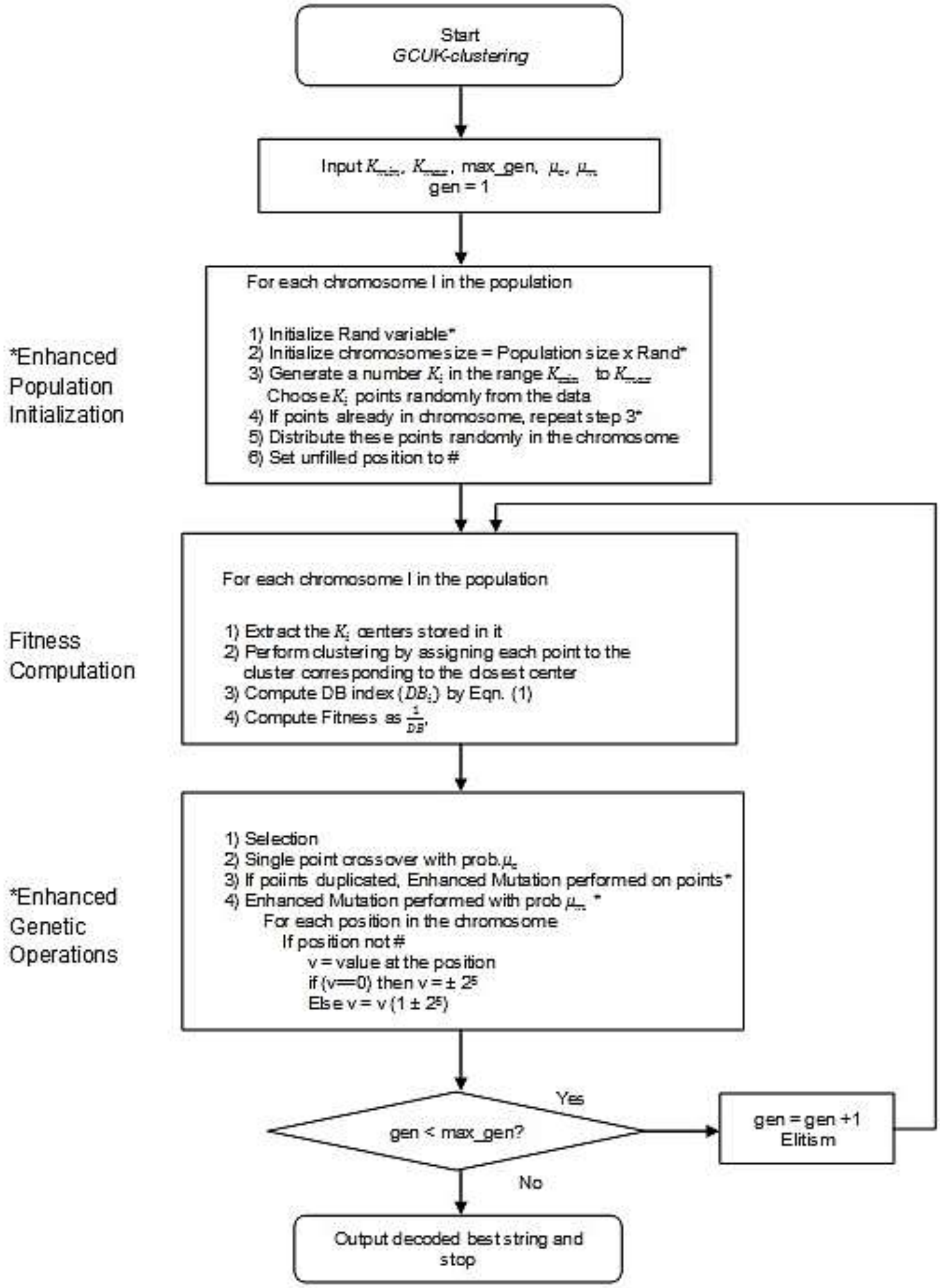

Fig. 2. Flowchart of the enhanced GCUK algorithm

It was argued that the random selection and the random number of genes in a starting population can result in less beneficial genes and chromosomes. Therefore, it was required to implement the modifications depicted in Figure 3 (right). Enhanced Population initialization, the first enhancement, tries to address the issue where not all services were clustered by GCUK (partial clustering). To ensure all services are clustered, a bigger chromosomal size is necessary. To expand the size of the chromosome, it is advised to multiply the present population size by the Rand value. Eq. (1) is the new size of the chromosomes,

$$\text{Chromosome size} = \text{Population size} \times \text{Rand} \tag{1}$$

```
/*Enhanced_Population_initialization
*/

Initialize population size
(PopulationSize)

Initialize Rand variable

Initialize chromosome size =
PopulationSize x Rand


Initialize minimum and maximum K
values


/*Main initialization loop          */

repeat

  Initialize chromosome with random
  services no.

      If service already in chromosome

      Then choose another service

until the last number of Population


/*End of
Enhanced_Population_initialization */

/*Enhanced_Genetic_operations*/
```

```
/*Population_initialization
*/

Initialize population size
(PopulationSize)

Initialize chromosome size =
PopulationSize


Initialize minimum and maximum
K values


/*Main initialization loop
*/

repeat

  Initialize chromosome with
  random service no.

until the last number of
Population


/*End of
```

Fig. 3. The original GCUK (left) and enhanced GCUK (right) pseudocodes

The value of Rand can be chosen at random. Population size equals the number of services. A further improvement to Enhanced Population initialization is a condition that determines if the randomly picked services to be initialized into the chromosome are duplicates. Any duplicated service will be replaced by a service that has not been selected. The second upgrade, Enhanced Genetic operation, tries to prevent services that emerge from crossover and mutation operations from duplicating. The suggested innovation substitutes duplicate services in the chromosome caused by crossover and mutation procedures automatically. These enhancements are essential for preventing service overlap and obtaining comprehensive clustering solutions.

3. Results

3.1 GCUK Preliminary Experiment Results

The chromosomes of the initial population generated by the GCUK based on the implementation using the heterogeneous cloud service dataset are shown in Table 1.

Table 1
The initial population created by the GCUK algorithm

Chromosome	Clusters	Services	Fitness	Chromosome	Clusters	Services	Fitness
C1	20	34	3.03	C39	3	74	1.93
C2	17	21	3.04	C40	20	43	1.41
C3	4	5	28.86	C41	6	6	0.00
C4	9	9	0.00	C42	21	37	1.55
C5	17	44	2.11	C43	10	63	0.81
C6	13	61	1.08	C44	13	14	10.12
C7	15	58	1.35	C45	22	27	11.06
C8	17	25	6.18	C46	19	55	1.41
C9	20	43	1.65	C47	18	32	2.53
C10	18	29	2.35	C48	14	55	1.09
C11	16	30	1.89	C49	18	37	2.20
C12	5	72	1.19	C50	11	15	2.25
C13	15	24	1.90	C51	7	67	1.04
C14	17	56	0.72	C52	10	11	7.05
C15	7	7	0.00	C53	14	62	1.23
C16	5	6	8.98	C54	11	66	0.95
C17	9	67	1.03	C55	11	64	0.71
C18	8	10	4.07	C56	8	9	5.81
C19	17	45	1.11	C57	12	18	1.38
C20	13	22	2.30	C58	17	53	1.26
C21	2	75	2.80	C59	12	64	0.62
C22	13	15	6.46	C60	16	40	1.44
C23	18	46	2.63	C61	16	43	1.19
C24	12	60	0.91	C62	14	56	1.29
C25	4	5	5.16	C63	17	29	2.28
C26	7	10	1.54	C64	17	39	1.40
C27	11	64	1.01	C65	17	33	1.06
C28	11	66	0.98	C66	10	63	1.30
C29	13	53	0.94	C67	20	41	1.46
C30	14	58	0.91	C68	10	65	1.07
C31	19	49	1.67	C69	12	64	1.13
C32	15	32	1.62	C70	20	22	7.51
C33	16	58	1.00	C71	19	31	2.42
C34	13	56	0.89	C72	12	14	9.96
C35	18	51	1.63	C73	15	49	1.41
C36	19	46	1.47	C74	20	42	1.68
C37	17	46	1.24	C75	21	29	3.61
C38	21	28	3.06	C76	5	6	8.22

Table 1 displays the initial population generated by the initialization function of the GCUK algorithm. It indicates that the number of cloud services in column 3 is not optimal, which may result in insufficient clustering. The actual ideal cloud service number is 76 (maximum number of services). Figure 4 depicts a solution of a chromosome derived *via* GCUK clustering.

#	2	58	#	#	18	#	#	76	#	#	#	68

Fig. 4. The solution produced by the GCUK automatic clustering on a heterogeneous cloud service dataset

Figure 4 depicts the final chromosome, which is the optimal solution with the maximum fitness developed by the GCUK. This chromosome is comprised of four clusters and five services. The answer is the C3 chromosome in Table 1. The chromosome has been identified as the local optimum or an early determination that it is the best option. Unfit clusters are distributed by a service per cluster, with one cluster including multiple services. Considering the results, the solution's number of services indicates partial clustering. Figure 5 displays the results of ten GCUK simulations.

1st run: 132ms

#	35	#	17	#	18	#	65	#	73	#	28	#	70	74	63	#	71	#	19

2nd run: 153ms

#	46	#	58	#	41	#	61	#	34	47	#	#	#	#	#	#	#	#	#

3rd run: 142ms

#	37	#	59	#	42	#	63	69	#	64	#	61	#	7	#	41	#	32	#

4th run: 130ms

#	63	#	38	#	47	#	28	#	55	#	29	#	21	8	#	#	63	#	38

5th run: 154ms

#	26	23	#	72	#	49	#	52	#	60	#	56	#	53	#	8	#	24	#

6th run: 148ms

#	51	#	37	#	11	#	75	#	33	#	69	#	39	#	64	#	29	43	#

7th run: 154ms

#	72	#	41	#	16	#	40	#	25	#	17	#	3	#	5	#	38	32	#

8th run: 132ms

#	24	#	7	#	16	#	76	70	#	26	#	20	#	14	#	5	#	39	#

9th run: 142ms

#	67	#	6	#	74	#	17	#	63	#	11	#	34	9	#	48	#	54	#

10th run: 109ms

#	42	47	#	7	#	54	#	67	#	34	#	#	42	47	#	#	#	#	#

Fig. 5. The solutions produced from GCUK 10 run on a heterogeneous cloud service dataset

Figure 5 demonstrates that all solutions are suboptimal and contain unsuitable clusters. In addition, the number of clustered services was less than 76, indicating just partial clustering. The average execution time of 10 iterations was 139.6 milliseconds. Even though it was suggested that the GCUK was adaptable in its handling of various types of data sets, it failed to effectively cluster the heterogeneous cloud service dataset, resulting in overlapping and incomplete solutions. In the meantime, the GCUK's DB Index created solutions that are caught in a local optimum. Time-wise, the GCUK is ideal for the automatic clustering of the double auction because the time required to provide solutions was less than one second. As a result, the GCUK algorithm for automatic clustering has been improved and the results will be reported in the following section.

3.2 Proposed Enhanced GCUK Experiment Results

The GCUK algorithm was enhanced in the population initialization and GA genetic operations. A GCUK function called Enhanced Population Initialization was proposed. The results of the proposed enhancement are shown in Table 2.

Table 2
Results of the proposed enhanced population initialization (left) compared to the standard population initialization of GCUK (right) with a 95% confidence level

Length of chromosome	Enhanced population initialization				Standard GCUK population initialization			
	Number of clusters	Number of services	Solution fitness	Execution time (ms)	Number of clusters	Number of services	Solution fitness	Execution time (ms)
76	1	76	0	132	9	21	2.1737	153
	1	76	0	110	9	30	2.1694	148
	1	76	0	121	5	35	0.4914	142
	1	76	0	97.6	5	5	0	109
	1	76	0	107	5	7	2.3574	132
Average	1	76	0	113.52	6.6	19.6	1.43838	136.8

The increased population initialization (left) produced the greatest number of services (76 services) in the initial population, as shown in Table 2. In contrast, the conventional GCUK produced an average of only 19 services (incomplete clustering). Due to the finite size of the chromosome, only one cluster can be formed for enhanced population initialization. Consequently, solution fitness cannot be determined from a single created cluster. Figure 6 depicts a solution found through increased population initialization.

11	48	51	21	50	7	41	20	24	34	16	12	15	35
	46	23	43	42	28	65	27	38	32	25	53	73	8
	49	63	36	5	26	58	60	68	10	9	71	2	40
	3	56	1	44	57	19	29	18	14	61	22	72	76
	69	4	62	33	30	70	13	52	64	47	17	31	37
	59	6	54	75	45	74	39	66	55	67			

Fig. 6. The solution was obtained from the enhanced population initialization

In Figure 6, the solution comprised all cloud services (from No. 1 to 76). The length of the chromosome is fixed according to the population size, hence there is no space to separate the services into clusters using the '#' symbol. Figure 6 demonstrates that there is no service duplication in the proposed solution. The second improvement was the utilization of the Rand variable.

3.2.1 Rand variable utilization results

The second enhancement of the GCUK was done by utilizing the Rand variable to extend the chromosome size as formulated in Eq. (1). The Rand was initially tested in the range of [1,1.5], where

Rand = 1 is the standard GCUK chromosome size. The results of the experiments on the heterogeneous cloud service dataset for five runs are presented in Table 3 and Table 4.

Table 3
Results of the enhanced CGUK that utilized Rand from 1 (standard GCUK chromosome size) to 1.1 with a 95% confidence level

Rand	Number of clusters	Number of services	Highest Fitness	Execution Time (ms)
1	1	76	0	132
	1	76	0	110
	1	76	0	121
	1	76	0	97.6
	1	76	0	107
Average	1	76	0	113.52
1.025	3	76	37.7988	109
	3	76	22.0855	109
	3	76	12.2829	85.6
	3	76	32.3518	85.5
	3	76	61.5072	108
Average	3	76	33.2052	99.42
1.05	5	76	3.692	87.6
	5	76	60.1388	109
	5	76	5.1914	119
	5	76	17.8669	109
	5	76	4.3847	108
Average	5	76	18.2547	106.52
1.075	7	76	3.0665	108
	6	76	35.0442	106
	7	76	2.7758	109
	7	76	17.3462	109
	7	76	37.8604	97.6
Average	7	76	19.21862	105.92
1.1	8	76	7.609	107
	8	76	2.773	97.4
	7	76	3.772	87.5
	8	76	3.210	85.4
	8	76	4.050	108
Average	8	76	4.2828	97.06

Table 3 and Table 4 demonstrate that the clustering is comprehensive (maximum number of cloud services). This is a result of the improved population initialization that initializes all cloud services within the population. The enhancement's average execution time was comparable to that of the GCUK.

Table 3 and Table 4 demonstrate that as Rand values climbed, so did the number of clusters. The rise can be ascribed to the use of the Rand variable, which has expanded the size of chromosomes and made room for the GCUK to form more clusters. With the increased allocation of space, more clusters were produced. The results suggested that a higher chromosomal size led to the formation of more clusters, which could increase monitoring and management costs. Therefore, chromosome length must be optimized.

Table 3 and Table 4 additionally indicated that chromosomes with Rand values between 1.025 and 1.1 possess a high level of fitness. It indicates that the best Rand values are between 1.025 and 1.1. The explanation for this is that the chromosome with a Rand value of 1.025 is longer than the typical GCUK chromosome (Rand = 1), which yielded the lowest fitness. When utilizing a Rand value of up to 1.1, the chromosomal length is not excessive and yet produces good fitness. Therefore, the lower border and upper boundary of the Rand variable are set to 1.025 and 1.1, respectively. According to the results, the best Rand value is 1.025.

Table 4
Results of the enhanced CGUK that utilized Rand from 1.2 to 1.5 with a 95% confidence level

Rand	Number of clusters	Number of services	Highest Fitness	Execution Time (ms)
1.2	13	76	2.1437	98.4
	14	76	3.5632	97.6
	11	76	2.1229	120
	15	76	2.4627	108
	11	76	2.1796	129
Average	13	76	2.4944	110.6
1.3	16	76	2.0873	121
	19	76	2.0871	96.6
	20	76	2.8102	109
	18	76	2.6826	118
	18	76	2.7256	106
Average	18	76	2.4785	110.12
1.4	21	76	2.2280	164
	25	76	2.4511	108
	23	76	2.4660	119
	21	76	2.3250	97.6
	20	76	2.3929	118
Average	22	76	2.3726	121.32
1.5	27	76	2.7207	120
	27	76	3.2546	118
	27	76	3.1991	131
	27	76	2.7534	109
	27	76	3.3324	131
Average	27	76	3.0520	121.8

3.2.2 Enhanced genetic operations

The enhanced genetic operations automatically replace the overlapping services during crossover and mutation. The results of the enhanced genetic operations are shown in Table 5 and compared with the standard genetic operations results.

Based on the acquired results, the increased genetic operations where there are no overlapping services have effectively solved the overlapping services in the solutions. Due to the elimination of overlapping services, the average execution time of the enhanced GCUK when using the enhanced genetic operations has improved marginally but is not significantly different from the conventional GCUK.

The proposed improvements have allowed the improved GCUK to prevent overlap and partial solutions. When evaluated on a dataset of heterogeneous cloud service heterogeneity, the improved GCUK largely yielded unique and exhaustive clustering solutions. Exclusive clustering methods are essential for the clustering of diverse cloud services because they can divide the dataset into discrete, more homogeneous clusters before the start of a negotiation. Complete clustering solutions prevent cloud services from being regarded as noise or anomalies. For some instances in which the solution becomes incomplete after genetic processes, the increased fitness computations will be applied. The GCUK used the Davies-Bouldin (DB) Index to calculate chromosome fitness [1-3]. The upgraded GCUK is suited for the automatic clustering of the double auction because the execution time of the method was less than 1 second.

Table 5
The results of the enhanced genetic operations (left) and the standard GCUK genetic operations (right) with a 95% confidence level

Run	Enhanced genetic operations			Standard GCUK genetic operations		
	Number of services	Overlapping services in the solution	Execution time (ms)	Number of services	Overlapping services in the solution	Execution time (ms)
1st	76	0	107	21	2	153
2nd	75	0	98.7	30	2	148
3rd	76	0	109	35	3	142
4th	76	0	109	5	0	109
5th	74	0	98.8	7	0	132
6th	76	0	98.8	15	1	132
7th	76	0	97.7	22	3	123
8th	75	0	98.3	12	1	142
9th	76	0	87.8	8	0	112
10th	76	0	96.7	35	2	141
Average	76	0	100.18	19	1	133.4

4. Conclusions

Experimental Evaluation of the enhanced GCUK is presented in this paper. The enhanced GCUK results showed improvements compared to the original GCUK. The findings support the argument that the GA-based automatic clustering technique is still progressive and enhancement is required for multi-attribute optimization. In the future, experiments should be conducted to test the enhanced GCUK results dealing with mixed data sets with numerical and categorical features.

Acknowledgement
This research was funded by a grant from the Ministry of Higher Education of Malaysia (FRGS Grant Reference Code = FRGS/1/2020/ICT02/UPSI/02/2, Project Code = 2020-0208-109-02). The authors would like to thank Universiti Pendidikan Sultan Idris (UPSI) for supporting this study.

References
[1] Adnan, M. H., M. F. Hassan, I. A. Aziz, O. Nurika and M. S. Husain. "Modified ISR hyper-heuristic for tuning automatic genetic clustering chromosome size." In *IOP Conference Series: Materials Science and Engineering*, vol. 932, no. 1, p. 012065. IOP Publishing, 2020. https://doi.org/10.1088/1757-899X/932/1/012065
[2] Muhamad Adnan, Muhamad Hariz. "Double Auction-Based Negotiation Framework For Heterogeneous And Multi-Attributes Cloud Services." PhD diss., Universit Teknologi PETRONAS, 2019.
[3] Adnan, Muhamad Hariz Muhamad, Mohd Fadzil Hassan and Nurul Akhmal Mohd Zulkefli. "A framework of heterogeneous cloud service and multi attributes negotiation using double auction." *International Journal* 9, no. 3 (2020). https://doi.org/10.30534/ijatcse/2020/220932020
[4] Almanza-Ortega, Nelva Nely, Joaquín Pérez-Ortega, José Crispín Zavala-Díaz and José Solís-Romero. "Comparative analysis of K-means variants implemented in R." *Computación y Sistemas* 26, no. 1 (2022): 125-133. https://doi.org/10.13053/cys-26-1-4158
[5] Ariyaratne, M. K. A. and T. G. I. Fernando. "A comprehensive review of the firefly algorithms for data clustering." *Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems* (2022): 217-239. https://doi.org/10.1007/978-3-031-09835-2_12
[6] Azhir, Elham, Nima Jafari Navimipour, Mehdi Hosseinzadeh, Arash Sharifi and Aso Darwesh. "An automatic clustering technique for query plan recommendation." *Information Sciences* 545 (2021): 620-632. https://doi.org/10.1016/j.ins.2020.09.037

[7] Bandyopadhyay, Sanghamitra and Ujjwal Maulik. "Genetic clustering for automatic evolution of clusters and application to image classification." *Pattern recognition* 35, no. 6 (2002): 1197-1208. https://doi.org/10.1016/S0031-3203(01)00108-X
[8] Dey, Alokananda, Sandip Dey, Siddhartha Bhattacharyya, Jan Platos and Vaclav Snasel. "Quantum inspired meta-heuristic approaches for automatic clustering of colour images." *International journal of intelligent systems* 36, no. 9 (2021): 4852-4901. https://doi.org/10.1002/int.22494
[9] Eesa, Adel Sabry and Zeynep Orman. "A new clustering method based on the bio-inspired cuttlefish optimization algorithm." *Expert Systems* 37, no. 2 (2020): e12478. https://doi.org/10.1111/exsy.12478
[10] Adnan, Muhamad Hariz Muhamad, Mohd Fadzil Hassan, Izzatdin Abdul Aziz, Okta Nurika and Mohd Shahid Husain. "An Overview of Current Multi-Attribute Techniques in Double Auction Frameworks." *Turkish Journal of Computer and Mathematics Education Vol* 12, no. 3 (2021): 877-883. https://doi.org/10.17762/turcomat.v12i3.798
[11] Hashemi, Seyed Emadedin, Madjid Tavana and Maryam Bakhshi. "A new particle swarm optimization algorithm for optimizing big data clustering." *SN Computer Science* 3, no. 4 (2022): 311. https://doi.org/10.1007/s42979-022-01208-8
[12] José-García, Adán and Wilfrido Gómez-Flores. "Automatic clustering using nature-inspired metaheuristics: A survey." *Applied Soft Computing* 41 (2016): 192-213. https://doi.org/10.1016/j.asoc.2015.12.001
[13] Kapoor, Shubham, Irshad Zeya, Chirag Singhal and Satyasai Jagannath Nanda. "A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation." *Procedia computer science* 115 (2017): 415-422. https://doi.org/10.1016/j.procs.2017.09.100
[14] Kumar, Rishi, Mohd Fadzil Hassan and Muhamad Hariz M. Adnan. "A Principled Design of Intelligent Agent for the SLA negotiation process in cloud computing." In *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, pp. 383-387. IEEE, 2022. https://doi.org/10.1109/ICCIT52419.2022.9711663
[15] Kumar, Rishi, Mohd Fadzil Hassan and Muhamad Hariz M. Adnan. "Intelligent negotiation agent architecture for SLA negotiation process in cloud computing." In *Proceedings of International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020*, pp. 771-778. Springer Singapore, 2021. https://doi.org/10.1007/978-981-33-4087-9_62
[16] Kumar, Rishi, Mohd Fadzil Hassan and Muhamad Hariz M. Adnan. "Analysis of Intelligent Agent and Its Components for the SLA Negotiation Process in Cloud Computing." In *International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi Petronas, Malaysia*, pp. 547-552. Singapore: Springer Nature Singapore, 2022. https://doi.org/10.1007/978-981-16-2183-3_51
[17] Stańczak, Jarosław and Jan W. Owsiński. "Evolutionary k-Means Clustering Method with Controlled Number of Clusters Applied to Determine the Typology of Polish Municipalities." In *International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets*, pp. 436-446. Cham: Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-95929-6_33
[18] Li, Kangshun, Yong Liu and Wenxiang Wang, eds. *Exploration of Novel Intelligent Optimization Algorithms: 12th International Symposium, ISICA 2021, Guangzhou, China, November 20–21, 2021, Revised Selected Papers*. Springer Nature, 2022. https://doi.org/10.1007/978-981-19-4109-2
[19] Xu, Zhuang, Yue Yin, Haitao Chen, He Xu and Peng Li. "Algorithm for determining number of clusters based on dichotomy." In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pp. 180-185. IEEE, 2020. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00045
[20] Zhong, Xin and Frank Y. Shih. "Automatic Image Pixel Clustering based on Mussels Wandering Optimization." *International Journal of Pattern Recognition and Artificial Intelligence* 35, no. 02 (2021): 2154005. https://doi.org/10.1142/S0218001421540057