Original Article

Hybrid Data Fragmentation Using Genetic Killer Whale Optimization-Based Clustering Model

Zina Jabbar Ahmed[1]* And Saadi Thalli Alluhaibi[2]

*[1,2]Department of computer. College of computer science and mathematics, University of Tikrit, Tikrit, Iraq
[1]E-mail: zenia.j.ahmed35537@st.tu.edu.iq, [2]E-mail: Saadi.alluhaibi@tu.edu.iq

***corresponding Author:** Zina Jabbar Ahmed

Department of computer. College of computer science and mathematics, University of Tikrit, Tikrit, Iraq, [1]E-mail: zenia.j.ahmed35537@st.tu.edu.iq
DOI: 10.47750/pnr.2022.13.S02.39

Abstract

Effective storage structure is essential in distributed data processing systems as it determines the users' consistent, scalable and reliable data accessibility. A flexible distributed database system's major aspects are data fragmentation and subsequent allocation. An efficient data fragmentation technique must support both the horizontal and vertical categorization of the tuples so that the frequency and type of queries for the data statistics are learned along with the data pattern and the attributes that impact these patterns. However, most existing fragmentation methods are based only on the query frequency and type for the data statistics, leading to less effective analysis of the data patterns and attributes in large databases. To avoid the data loss and imperfect grouping of large databases, this paper presents a hybrid data fragmentation model using an advanced optimization-based clustering model for performing horizontal and vertical fragmentation. This proposed model, called Genetic Killer Whale Optimization based Clustering (GKWOC), is developed by integrating the Killer Whale Optimization (KWO) and Genetic Algorithms (GA) into the standard subspace clustering algorithm. This optimization-based hybrid fragmentation model performs fragmentation in terms of the tuples and the attributes. Since the databases can be high dimensional, subspace-based clustering can improvise good fragments for distributed databases and reduce the data access time for the users. This proposed model showed better fragmentation performance than the state-of-the-art methods during the evaluations, thus suitable for large distributed databases with competent access time and reduced error rates.

Keywords: Distributed databases, Data fragmentation, hybrid fragmentation, Genetic Killer Whale Optimization, subspace clustering

1. INTRODUCTION

Database Management Systems (DBMS) is one of the most efficient ways to improve the performance of applications that must analyze and process a large amount of data in a cloud environment [1]. Distributed DBMS (DDBMS) [2] have been largely utilized in sectors like Banking, Healthcare, Education, etc., as they replace the centralized administration and effectively process large data with minimum complexity and data loss. The distributed data must be transparent to the users and logically related so that the relationship between the different locations is defined based on the specified structural formations [3]. To formulate an ideal distributed database in the cloud environment, two main processes must be performed in the design stage of these distributed databases. They are fragmentation and allocation, which are NP-hard problems and require extensive analysis to solve [4]. Fragmentation aims to segment the global relation databases into fragment relations, while allocation aims to position these fragments in suitable sites or distribution locations. Fragmentation enhances the system performance and access locality by the users and applications. It reduces the amount of irrelevant data access by providing the necessary subsets of relations instead of the entire database [5]. The further updated data can be transferred to the previous sites of the corresponding subsets and replaced. Decomposition of a relation into fragments also enhances concurrent processing since parallel processes can simultaneously utilize the same subset or different subsets. With the ever-increasing volume of databases and continuous updates, it is difficult to determine the distribution strategy during the designing process since the data access must be fast and re-organization of data is frequent. In such scenarios, even the Hadoop file system aims to distribute the data while not considering the cohesiveness or data patterns [6]. As the locality access for the applications is defined on the subsets of the relational database, the number of access from and to the disk must be minimized so that the incoming queries are executed efficiently. This can be achieved with precision only when the databases are fragmented intelligently. Thus, fragmentation is vital in increasing the system throughput and decreasing the response time of accessing data.

Intelligent Fragmentation of the databases effectively establishes the cohesiveness between fragments and aligns the data to the users in perfect patterns. However, the traditional database fragmentation methods are insufficient as the latest distributed databases are large with numerous attributes and tuples [7]. The fragmentation methods are either horizontal or vertical fragmentation. Horizontal fragmentation is a row-wise operation, while vertical fragmentation is a column-wise operation [8]. While these methods have greater benefits individually, similarly, they possess individual drawbacks. Therefore, recent studies have focused on developing newer proficient fragmentation methods that remove the need for

irrelevant data access during the query execution and reduce the data transmission between the replication sites. The attribute-based fragmentation in vertical fragmentation is faster, but they neglect the cohesiveness. Similarly, the tuples-based fragmentation depends on similarity to determine the fragments, and the data pattern is not effectively determined without the attributes [9]. Considering these benefits and drawbacks, hybrid fragmentation is developed to complement these two methods. To attain this objective, some studies have utilised clustering methods [10]. In recent years, meta-heuristic-based optimization algorithms have been integrated to enhance the behaviours of traditional clustering methods for database fragmentation.

This paper presents an intelligent, hybrid database fragmentation technique using the GKWOC method, combining the KWO and GA with the standard subspace clustering algorithm. This optimization-based hybrid fragmentation model performs fragmentation in terms of the tuples and the attributes. This hybrid method performs horizontal and vertical fragmentation and can improvise good fragments for distributed databases with reduced data access time. The KWO and GA algorithms were selected based on their superior optimization performance and their similarity in forming groups or hierarchy orders among individuals to form the best solutions. This group formation behaviour is similar to the clustering and fragmentation strategies; hence, they are chosen over the other optimization algorithms. The rest of this research paper is built as- Recent prominent research studies are discussed briefly in Section 2. The proposed research contribution, GKWOC based fragmentation approach, is presented in Section 3. The experiments and comparison outcomes are presented in section 4, while section 5 completes this article.

2. RELATED WORKS

Database fragmentation is a basic and vital research topic among cloud-based researchers. Many studies have tried their best to develop efficient fragmentation methods by applying different strategies. Harikumar and Ramachandran [11] developed a Hybridized fragmentation model using the subspace clustering algorithm. Like the other clustering algorithms, this subspace clustering determines a similarity metric and group's similar data into clusters. The subspace clustering algorithm uses the projected clustering to determine the clusters and then partitions the data concerning the tuples and the attributes. However, this method has limitations in the generalization of the fragmentation problem. Amer et al. [12] proposed an optimized heuristic horizontal fragmentation and allocation approach, reducing the overall Transmission Costs (TC). This fragmentation model used predicates for the objective and heuristic allocation for integrating the non-replication cases with the site clustering algorithm to reduce the allocation costs. This model converts the Communication Cost Matrix (CSM) into the Distance Costs Matrix (DCM) so that the objective of transmission cost reduction is achieved. This model also achieved adaptive access frequency and improved the overall capacity. However, this model increased the computational complexity.

Rahimi et al. [13] proposed a Hierarchical vertical fragmentation model using the Modified Bond Energy Algorithm (MBEA). In this model, the BEA is modified by using an affinity measure to improve the cluster formation of the attributes. This fragmentation model simultaneously computes the cost of allocating each data fragment to the related site and selects the best site. This model reduced the overall computational complexity and the response time for accessing the data. However, this model has drawbacks in optimizing the cost for accessing the update frequency of fragments. Luong et al. [14] developed a fragmentation approach using the Ant Colony Optimization (ACO) technique. This approach performs vertical fragmentation using the ACO to optimize the clustering algorithm for large data spaces. This ACO-based fragmentation approach reduced the overall costs and computational complexity but has also increased the response time due to the randomness in the search process. Drissi et al. [15] proposed horizontal fragmentation for fuzzy querying databases using different query execution strategies. This method reduced the number of frequent access of the fragments and minimized response time. However, this method has drawbacks in handling larger distributed databases. Ge et al. [16] presented a randomly distributed fragmentation method using Distributed memetic algorithm (DMA). This method utilized the optimization algorithm with a dynamic grouping recombination operator and mutation operator. This fragmentation method has used splicing-driven local search (SDLS) to perform rare utility fragments into the population without violating constraints. This method increased the speedup ratio and reduced the makespan and response time.

Amer [17] proposed using the K-means clustering-based approach for fragmentation in DDBS. This method uses a heuristic k-means approach for the vertical fragmentation strategy and allocation of the fragments. Query Usage Matrix (QUM) is used to determine the suitability of the sites that release the queries, and a novel method is used to refine the fragments to avoid overlapping clusters. This K-means-based method reduces the data locality maximization and communication costs and improves the data cohesion between the fragments when allocating. However, this method does not study the optimality problem while clustering the databases. Ge et al. [18] presented Set-based Adaptive Distributed Differential Evolution (S-ADDE) for fragmentation with anonymity assurance. In this model, the concept of k-anonymity has been utilized to provide anonymity-based fragmentation through the S-ADDE algorithm. This S-ADDE model incorporates the island model to define the population diversity and then integrates set-based mutation and crossover operators to modify the continuous domain to a discrete domain through evolution. This hybrid model reduced privacy issues and improved response time effectiveness. Yet, this model has utility issues in terms of communication cost. Ge et al. [19] introduced a Knowledge Transfer-based DDE (KT-DDE), which reduces communication costs and improvises privacy preservation in fragmenting the databases. In this model, the fragmentation information is exchanged between the optimizers, and then the trial individuals are generated to determine the competitive trials which are the optimal

fragments. This model increased the solution accuracy and improved convergence speed and computational efficiency. However, the speedup ratio of this model is not balanced, leading to uneven processing time.

Mehta et al. [20] developed Differential bond energy (DBE) algorithm based on optimal vertical fragmentation. This model utilized the DBE algorithm for determining the optimal partition point based on the global affinity measure (GAM) value. This model improves the vertical fragmentation effectiveness even for high-dimensional data problems. However, this method might increase the computational complexity when the complex databases are needed to be fragmented. Ge et al. [21] introduced a multitasking distributed differential evolution (MDDE) algorithm to improve database fragmentation with high privacy preservation for the users. This algorithm utilizes sequential fragmentation in which the redundant computations are reduced through similarity and perturbation-based operators for multitasking distributed fragmentation with effective allocation of sensitive attributes into separate data fragments. This method increased the performance with better solution accuracy, convergence speed, and scalability. Castro-Medina et al. [22] defined a new Dynamic Horizontal Fragmentation through Content-Based Queries. This method used content-based queries to provide better-performing schemas for dynamic fragmentation. It improved response times by 98.84% and reduced irrelevant access to irrelevant tuples. However, this method increased the computational complexity.

From the above-discussed studies in the literature, the major limitations are that most of these methods are based only on the query frequency and type for the data statistics, leading to less effective analysis of the data patterns and attributes in large databases. Similarly, the frequency of queries, affinity or predicates require sufficient empirical data for performing optimal fragmentation but are unavailable during the initial stage due to periodic database updates. The other problem is that the predicates often need to be changed manually to determine the frequency threshold leading to inconveniences in the big data problems. This paper has presented a hybridized fragmentation method using the GKWOC algorithm to avoid such problems.

3. METHODOLOGY

The proposed method focuses on improving the fragmentation performance by modelling the database fragmentation into a clustering problem and resolving it using an optimization-based clustering algorithm. This method is aimed at improvising the database replication for flexible query handling. The proposed method incorporates the hybrid or mixed fragmentation strategy to achieve this objective by combining the horizontal and vertical partitioning methods. As the subsets of relations determine the access locality of the applications, the fragmentation method has the liberty to follow row-wise, column-wise or both strategies for fragmenting the databases. While the individual horizontal and vertical fragmentation can provide either disjoint instances of relations or relations based on the subset of attributes, the hybrid method can provide fragments with subsets of attributes and instances. The hybrid fragmentation method divides the relation database into many arbitrary blocks based on the data instances' usage frequency and the queries' attributes. The generated fragments are allocated to specific sites based on parameters such as site location, accessibility and capacity factors. The challenging task is the determination of the fragments' attributes and the instances since these fragments must ensure the completeness of the attributes and the instances with reduced computation and communication costs.

3.1. Genetic Killer Whale Optimization

GKWO is formulated by utilizing the mutation and crossover operators of the genetic algorithm into the standard KWO algorithm [23]. KWO is based on the killer whale species' leadership hierarchy and hunting behaviour. The killer whales are classified as either Fish-Feeding Residents or Mammal-Hunting Transients based on the type of hunting strategy they perform. Fish-Feeding Residents live and hunt their prey in the same area while Mammal-Hunting Transients hunt the prey during migration seasons. These characteristics are passed over the generations through the memorizing capacity of the *Matriline* (mother line). These unique characteristics of the killer whales are mathematically modelled as the steps of the KWO. The algorithm initialization process is done by determining the initial parameters. The number of *Matriline,* global optimum solution, dimensions of the objective function, upper and lower bounds of search space and optimized variables, number of search space clusters and the iterations are initialized. Then the search space is clustered into K number of initial clusters to reduce the complexity of the exploration process.

For the NP-hard optimization problem, the optimization procedure will aim to find the optimal values for the parameters within the boundary of $[-x, x]$ of the N-dimensional search space. The GKWO algorithm clusters the range of optimization, and the group of search agents becomes the range for selecting the optimal solution. Then the population of the *Matriline* will be divided into groups of search agents based on the fitness values, with each group foraging a cluster in the search space. For each group of search agents, their role must be defined as either leader or member. Each search agent will be employed from the cluster's centroid in the process of finding the optimal value (local best) within the cluster. The clustering of the search space utilizes the matrix input of M points and K initial cluster centres. The total number of points in cluster C can be defined as$NP(C)$. $D(I, C)$, Denoting the Euclidean distance between point I and cluster C, the primary process of GKWO is to search for a partition K which is the local best within-cluster sum of squares by transferring the points from one cluster to another.

The clustering process of the optimization search space is based on the k-means clustering algorithm. Initially, each point $I(I = 1,2, \dots, M)$ identifies their closest and next-best closest clusters, namely $IC1(I)$ and $IC2(I)$ and the point I is defined to the cluster$IC1(I)$. The centre of the clusters is updated to the mean value of the points in that cluster. At the initial stage, all the clusters are assigned live cluster sets. Then, the optimal transfer stage is utilized. In this stage, each point $I(I = 1,2, \dots, M)$ is assigned to the clusters. If the clusters $C(C = 1,2, \dots, K)$ are updated in the last quick-transfer stage, the clusters will be included in the current set. Otherwise, they are left in the archive set until the end of the optimal transfer stage. Now, the point I is located in cluster $IC1(I)$, and therefore it must perform either of the two steps.

In the first case, it will compute the quantity of minimum using the formula$R2 = \frac{[NP(C)*D(I,C)^2]}{[NP(C)+1]}$, then the overall clusters $C(C \neq C1, C = 1,2, \dots, K)$ were clustered with the smallest$R2$. The value of $\frac{[NP(C)*D(I,C)^2]}{[NP(C)+1]}$ is memorized and will be the same until the cluster is renewed. If this value becomes greater than or equal to$\frac{[NP(C)*D(I,C)^2]}{[NP(C)+1]}$, then the reallocation will not be required, and $C2$ will become the new$IC2(I)$. If not, the point I will be assigned to cluster $C2$ and $C1$ will become the updated$IC2(I)$. The cluster centres will be renewed to be the average of the points assigned to the clusters and the two clusters in the transfer of point I will now be included in the live set. In the second case, the same process will be performed, but the minimum $R2$ will be computed over the cluster in the live set individually.

In the next step, the quick transfer will be performed by considering each point $I(I = 1,2, \dots, M)$ and putting $C1 = IC1(I)$ and$C2 = IC2(I)$. If the clusters $C1$ and $C2$ have the same values in the M steps, then point I is not needed to be validated. Computing the values of $R1 = \frac{[NP(C1)*D(I,C1)^2]}{[NP(C1)+1]}$ and $R2 = \frac{[NP(C2)*D(I,C2)^2]}{[NP(C2)+1]}$, if $R1 < R2$, then the point I will be allocated to $C1$ and if $R1 > R2$, then $IC1(I)$ and $IC2(I)$ will be archived and the cluster centres of $C1$ and $C2$ will be renewed. The final area of these two clusters will be used as the initial search space.

The killer whales will forage into these clusters, with each cluster having its specialized leader and members nominated from the group of search agents initially divided from the population of the *Matriline.* The echolocation of killer whales is mathematically modelled to represent the prey foraging behaviour for finding the best solution based on the objective function. Considering the depth of the prey and the killer whales' sonar denoted as d_F and d_o, respectively, SR denotes the slant range between the prey and killer whale, X denotes the horizontal range, and θ denotes the angle between the slant and horizontal ranges. This angle θ is computed as

$$\theta = \sin^{-1}\left(\frac{d_F - d_o}{SR}\right) = \tan^{-1}\left(\frac{d_F - d_o}{x}\right) \quad (1)$$

The velocity of the killer whale movements is computed to update their current positions in the direction of the prey location. The mathematical model for denoting the movements of the killer whales is represented as

$$\overrightarrow{v_i} \leftarrow \overrightarrow{v_i} + \vec{U}(0, \phi_1) \otimes (\overrightarrow{p_i} - \overrightarrow{x_i}) + \vec{U}(0, \phi_2) \otimes (\overrightarrow{p_g} - \overrightarrow{x_i}) \quad (2)$$

$$\overrightarrow{x_i} \leftarrow \overrightarrow{x_i} + \overrightarrow{v_i}.t \quad (3)$$

Eq. (2) is the velocity equation, and Eq. (3) denotes the location update equation. The current position of whales is considered as a set of coordinates describing a point in the search and is denoted as $\overrightarrow{x_i}$, the previous best position as $\overrightarrow{p_i}$, the velocity as $\overrightarrow{v_i}$ and time is denoted as t. If the $\overrightarrow{x_i}$ is better than the $\overrightarrow{p_i}$, then the current position will be represented as the previous best position in the next iteration. The value of the best objective function in the previous iteration will be denoted as $\overrightarrow{pbest_i} = \overrightarrow{p_g}$ and it will be compared with the next iteration value to obtain a better position for the whales. The positions of the whales are validated by using genetic operators. The $\overrightarrow{pbest_i}$ and the second-best $\overrightarrow{pbest_i}$ Whales are selected from the solution set to serve as the parent solutions. These two solutions are applied in the two-point crossover operation to formulate two new solutions and added to the larger solution set. These two solutions will resemble the parent solutions; hence, the 20% mutation operation is applied to enhance the solutions. Now, the solutions are compared in each iteration with the new solutions, and the optimal solution is selected. These crossover and mutation operators enhance the exploitation phase and lead to highly improved solutions. The new positions will be updated using $\overrightarrow{v_i}$ and the final best position among all iterations will be termed as $\overrightarrow{gbest_i}$, which returns the best (optimal) solution. The overall process involved in the proposed GKWO algorithm is summarized in Algorithm 1.

Algorithm 1: GKWO Algorithm

Input: W*= Initial best search agent, θ and all parameters
T= maximum number of iterations
Output: $\overrightarrow{gbest_i}$= Optimal search agent
Begin
Initialize the population of *Matriline* population $W = W_i(i = 1,2, \dots, n)$
Group the whole population into multiple groups

Cluster the search space using the k-means clustering process
For each cluster
Assign a group of *Matriline* randomly
Compute fitness for each individual in the groups
Select the leader based on better fitness
Assign other individuals as members
All members and the leader scan for prey
Leader decides the best location for prey
Members follow the leaders' orders
Leader re-scans for potential new sources of prey
If new prey > old prey location
Leader redirects the members to the new location
Members attack the prey in the new location
Else
Members attack the prey in the old location
End if
Return $\overrightarrow{p_g}$
End for
Update velocity and location using Eq. (2), (3)
Select $\overrightarrow{p_g}$ and $\overrightarrow{p_i}$ as parent solutions
Apply two-point crossover to parent solutions $\overrightarrow{p_g}$ and $\overrightarrow{p_i}$
Generate offspring solutions
Apply 20% mutation
Generate new solutions $\overrightarrow{p_j}$ and $\overrightarrow{p_k}$ by incorporating $\overrightarrow{p_g}$ and $\overrightarrow{p_i}$
Add $\overrightarrow{p_j}$ and $\overrightarrow{p_k}$ to the solution set
If either $\overrightarrow{p_g} < \overrightarrow{p_j}$ or $\overrightarrow{p_g} < \overrightarrow{p_k}$
Replace $\overrightarrow{p_g}$ with $\overrightarrow{p_j}$ and $\overrightarrow{p_k}$
Else
Retain $\overrightarrow{pbest_i} = \overrightarrow{p_g}$
Update velocity and location using Eq. (2), (3)
End if
Increment iteration by 1
Return all $\overrightarrow{pbest_i}$ from all iterations
Until iterations reach maximum T
Determine $\overrightarrow{gbest_i}$
Return $\overrightarrow{gbest_i} = W$ as the optimal solution
End

3.2. Clustering process

The proposed hybrid fragmentation is designed using the selection and the projection operators of the clustering method of relational databases as: $\sigma_p(\pi(t_1, t_2, \dots, t_t)R)$ where $t_1, t_2, \dots, t_t$ are the data instances (tuples), R denotes the records in the relational database, π denotes the selection operator and σ_p denotes the projection operator. The projected cluster is a subset of R_k records with A_k attributes so that the records in R_k are tightly grouped in the subspace of A_k. Applying the subspace clustering process, the bond energy is used for attribute clustering of bonded global attributes (Bong) [24]. Let D denote the database, $R = (r_1, r_2, \dots, r_N)$ be the set of records with each r_i described by the set of attributes $A = (t_1, t_2, \dots, t_t)$. During clustering, each point of the cluster is a record r_i and each dimension is an attribute. The cluster's centroid is mathematically computed as the algebraic average of all records along each attribute in the cluster. Medoids m_k is a record that is very closer to the centroid, and each cluster in a numerous clusters-based dataset is represented by a m_k to differentiate the clusters [25].

In general subspace clustering, similarity measures such as Manhattan distance or Euclidean distance measure are computed to determine the m_k and its locality L_k. However, this process is time-consuming, and hence the GKWO is applied in this stage to compute the m_k and the L_k. Then it will be easier to find all the records that fall within the distance of m_k. For this purpose, the GKWO utilizes the Euclidean distance measure and optimally selects the m_k and L_k. The application of GKWO reduces the complexity of the computation process.

Similarly, the clustering process of subspace clustering travels in the direction determined by the cardinality of the neighbourhood. This direction can be either horizontal or vertical. The GKWO is also applied to determine the optimal direction at each stage of the clustering process. This optimal direction process reduces the complexity of modifying the direction based on cardinality. Therefore, the inclusion of GKWO enhances the functioning of the standard subspace clustering. This GKWO-based clustering model forms the basis of the proposed hybrid fragmentation.

3.3. Fragmentation using GKWOC

The primary objective of the proposed study is to develop the fragmentation model that enables horizontal and vertical division of the overall relational tables of the database. Additionally, the fragmented sub-databases must be cohesive so that they can be efficiently distributed across different server locations and retrieved without any loss of information. The proposed hybridized algorithm focuses on partitioning a database so that the tuples are grouped based on the relevant attributes. The perfect fragmentation method must ensure the fragments are formed with cohesive tuples and corresponding attributes, making the data retrieval more cohesive.
Once the needed number of clusters is formed in the GKWOC method based on the optimized subspace clustering, the fragments must be formed from these clusters. It is possible that sometimes few attributes might not be assigned to any cluster since they are less relevant to the generated clusters. In similar scenarios, vertical partitioning of all these missed attributes is done corresponding to the tuples. When the necessary number of partitions is equal to the number of clusters, the clusters can be assigned to suitable sites based on user requests or application entries. But, if the available number of site locations is less than the available number of clusters, then the suitable clusters must be merged to limit the number of partitions so that each fragment is assigned to each of the initialized site locations. The merging process continues until the number of clusters equals the number of sites, but this process does not include the vertical partitioned missed attributes. Merging must be done based on a suitable similarity measure so that the merged cluster does not have irrelevant points. This study employs the cosine similarity measure to determine the similarity between two closely resembling clusters.

Compared to other measures such as the Jaccard index, Silhouette index, etc., the cosine similarity measure is sensitive to overlapping data. This feature of cosine similarity reduces the possibility of overlapping in the fragments. The complexity in computing the cosine similarity is lesser and provides better merging of the similar clusters. Cosine similarity between two clusters X and Y is computed as

$$\cos(A, B) = \frac{A.B}{\|A\|\|B\|} \quad (4)$$

The proposed fragmentation consists of two stages, namely Cluster formation using GKWOC and Hierarchical merging of the clusters. The second stage is optional, depending on the number of clusters and available sites. The cluster formation process is detailed in the previous two subsections. Algorithm 2 provides the tasks of fragmentation in a simpler process.

Algorithm 2: Proposed Hybrid Fragmentation

Input: Data tuples, Set K=0
Begin
Determine K clusters for $R(t_1, t_2, \ldots., t_t)$
Apply GKWOC
Form K clusters based on optimal medoids and locality
Form vertical partitions of unassigned attributes to V clusters
Let $m = (m_1, m_2, \ldots, m_k)$ bet set of medoids from k clusters
Initialize $M = K$
If $M = K$, then
    Assign K clusters to M sites
Else if $M \leq K$
    Apply Merging
    For $(i = 1\ to\ k)$
        For $(j = i + 1\ to\ k)$
            Compute Cosine Similarity using Eq. (4)
            Find the maximum value of $\cos(A, B)$
            Merge A and B
            Remove corresponding m_i from m
            $M = M - 1$;
        End for
    End for
Return M
End

4. EXPERIMENTS AND RESULTS

The proposed GKWOC-based hybrid fragmentation method is evaluated using two real datasets, project management and a hierarchical employee details database. These two datasets are available publicly in the UCI repository. The proposed method is implemented using Java programming codes and executed in Netbeans IDE. The performance is evaluated by executing different queries on the clustered and random fragments. The number of attributes in these datasets is 17, and the tuples are duplicated randomly to form 2000 samples. The clusters and the corresponding records generated using the proposed GKWOC method are shown in Table 1.

Table.1. Clusters with relevant attributes using GKWOC

Cluster id	Attributes	No. of records
C1	d_no, name, dept, salary	234

C2	F_name, salary, position	211
C3	P_no, P_name, Budget	246
C4	d_no, loc, Budget	375
C5	d_no, name, dept, Budget	166
C6	d_no, loc	287
C7	d_no, name, dept, salary	393
C8	L_name, site, position	88

The clusters are merged using the hierarchical merging process to formulate the fragments, as shown in Table 2.

Table.2. Fragments and Clusters with relevant attributes using GKWOC

Fragment id	Clusters Merged	No. of records
Frag1	C1,C2, C7	838
Frag2	C5,C6, C8	541
Frag3	C3, C4	621

The time taken to process different queries in random and clustered fragments using the proposed model is compared to evaluate the efficiency. Fig 1 shows the time comparison.

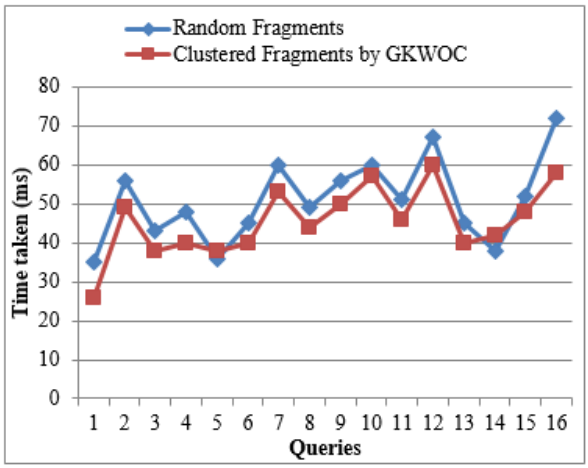

Fig.1. Time taken to execute different queries

From Fig.1, the time taken for executing different queries in the clustered fragments is less than the time taken by random fragments. This is because of the higher efficiency in clustering the databases with optimal direction. The execution time can be higher in certain cases due to the connection being established to the remote databases. Similar to the time, the number of fragments accessed for different queries in random and clustered fragments is sown in Fig 2.

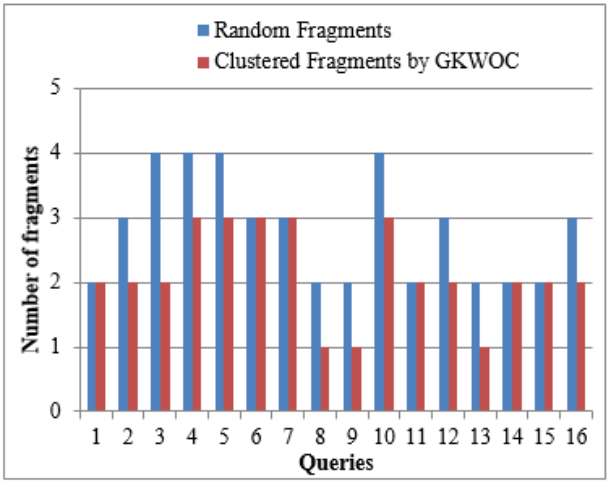

Fig.2. Number of fragments accessed for different queries

From Fig 2, it is seen that the number of fragments accessed by the applications or users during a certain time in clustered fragments is quite lesser or equal to that accessed in random fragments. This optimal performance of the proposed GKWOC-based fragmentation method is sufficient for performing real cloud database fragmentation and corresponding allocations.

The performance of the proposed GKWOC-based method is compared against the well-known existing methods in the literature. The Final Prediction Error (FPE) and execution time are the parameters used for this comparison.

FPE: FPE estimates the model-fitting error when the used model to predict new outputs. For the FPE, an optimal model is the one that minimizes the following equation:

$$FPE = SSE\left[1 + \frac{2p}{N-p}\right] \quad (5)$$

Here SSE is an index related to the prediction error or the residual sum of squares, N is the number of fragments, and p defines the number of parameters in the model.

Execution time: The total time taken to complete the designated fragmentation set using the proposed model. Execution time includes the time for loading the dataset and setting up memory for each step and subsequent supplementary processes.

For comparison, Subspace clustering [11], optimized heuristic horizontal fragmentation [12], MBEA [13], ACO [14], Fuzzy querying [15], DMA [16], K-means clustering [17], S-ADDE [18], KT-DDE [19], DBE [20], MDDE [21] and Dynamic Horizontal Fragmentation [22] are used. Table 3 shows the comparison results obtained for 100 queries.

Table.3. Performance comparison

Reference	Method	FPE (ratio)	Execution time (seconds)
[11]	Subspace clustering	0.05	2.865
[12]	Optimized heuristic horizontal fragmentation	0.11	4.570
[13]	MBEA	0.087	2.56
[14]	ACO	0.102	3.98
[15]	Fuzzy querying	0.265	4.11
[16]	DMA	0.165	5.65
[17]	K-means clustering	0.094	3.33
[18]	S-ADDE	0.087	4.67
[19]	KT-DDE	0.134	4.71
[20]	DBE	0.115	2.26
[21]	MDDE	0.107	4.08
[22]	Dynamic Horizontal Fragmentation	0.345	3.14
-	Proposed GKWOC	0.034	2.03

Table 3 shows that the proposed GKWOC-based fragmentation method has achieved better results than the methods in the literature. GKWOC has consumed less execution time and also reduced the overall error rates.

5. CONCLUSION

This paper aims to develop an efficient fragmentation method that can inherit the benefits of vertical and horizontal partitioning. To avoid the data loss and imperfect grouping of large databases, a hybrid data fragmentation model using Genetic Killer Whale Optimization based Clustering was developed by optimizing the standard subspace clustering algorithm. This proposed model showed better fragmentation performance than the state-of-the-art methods during the evaluations, thus suitable for large distributed databases with competent access time and reduced error rates. In future, more generalizations of this model will be investigated for adapting to different types of databases. Also, the feasibility of integrating this proposed method with better allocation schemes will be examined.

REFERENCES

1. Gelogo, Y. E., & Lee, S. (2012). Database management system as a cloud service. International Journal of Future Generation Communication and Networking, 5(2), 71-76.
2. Kamal, J. M. M., & Murshed, M. (2014). Distributed database management systems: architectural design choices for the cloud. In Cloud Computing (pp. 23-50). Springer, Cham.
3. Rana, M. S., Sohel, M. K., & Arman, M. S. (2018). Distributed Database Problems Approaches and Solutions-A Study. International Journal of Machine Learning and Computing (IJMLC).
4. Iacob, N. (2011). Fragmentation and data allocation in distributed environments. Annals of the University of Craiova-Mathematics and Computer Science Series, 38(3), 76-83.
5. Bhuyar, R. P., & Gawande, A. D. (2012). Distributed Database: Fragmentation and Allocation. Journal of Data Mining and Knowledge Discovery, 3(1), 58.
6. Shaha, T. R., Akhtar, M. N., Johora, F. T., Hossain, M. Z., Rahman, M., & Ahmad, R. B. (2019). A noble approach to develop dynamically scalable namenode in Hadoop distributed file system using secondary storage. Indonesian Journal of Electrical Engineering and Computer Science, 13(2), 729-736.
7. Castro-Medina, F., Rodríguez-Mazahua, L., Abud-Figueroa, M. A., Romero-Torres, C., Reyes-Hernández, L. Á., & Alor-Hernández, G. (2019). Application of data fragmentation and replication methods in the cloud: a review. In 2019 international conference on electronics, communications and computers (CONIELECOMP) (pp. 47-54). IEEE.

8. Bhardwaj, A., Singh, A., Kaur, P., & Singh, B. (2012). Role of Fragmentation in Distributed database system. International Journal of Networking & Parallel Computing, 1(1).
9. Fauzi, A. A. C., Rahman, W. F. W. A., Fauzi, A., & Weigelt, F. (2021). Managing Fragmented Database in Distributed Database Environment. Journal of Mathematics & Computing Science, 7(1), 8-14.
10. Kaundal, G., Kaur, S., & Vashisht, S. (2014). Review on Fragmentation in Distributed Database Environment. IOSR Journal of Engineering, 4(3), 28-32.
11. Harikumar, S., & Ramachandran, R. (2015). Hybridized fragmentation of very large databases using clustering. 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES) (pp. 1-5). IEEE.
12. Amer, A. A., Sewisy, A. A., & Elgendy, T. M. (2017). An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). Heliyon, 3(12), e00487.
13. Rahimi, H., Parand, F. A., & Riahi, D. (2018). Hierarchical simultaneous vertical fragmentation and allocation using modified Bond Energy Algorithm in distributed databases. Applied computing and informatics, 14(2), 127-133.
14. Luong, V. N., Solanki, V. K., & Cuong, N. H. H. (2019). Fragmentation in Distributed Database Design Based on Ant Colony Optimization Technique. International Journal of Information Retrieval Research (IJIRR), 9(2), 28-37.
15. Drissi, A., Nait-Bahloul, S., Benouaret, K., & Benslimane, D. (2019). Horizontal fragmentation for fuzzy querying databases. Distributed and Parallel Databases, 37(3), 441-468.
16. Ge, Y. F., Yu, W. J., Cao, J., Wang, H., Zhan, Z. H., Zhang, Y., & Zhang, J. (2020). Distributed memetic algorithm for outsourced database fragmentation. IEEE Transactions on Cybernetics, 51(10), 4808-4821.
17. Amer, A. A. (2020). On K-means clustering-based approach for DDBSs design. Journal of Big Data, 7(1), 1-31.
18. Ge, Y. F., Cao, J., Wang, H., Chen, Z., & Zhang, Y. (2021). Set-Based Adaptive Distributed Differential Evolution for Anonymity-Driven Database Fragmentation. Data Science and Engineering, 6(4), 380-391.
19. Ge, Y. F., Orlowska, M., Cao, J., Wang, H., & Zhang, Y. (2021). Knowledge transfer-based distributed differential evolution for dynamic database fragmentation. Knowledge-Based Systems, 229, 107325.
20. Mehta, S., Agarwal, P., Shrivastava, P., & Barlawala, J. (2022). Differential bond energy algorithm for optimal vertical fragmentation of distributed databases. Journal of King Saud University-Computer and Information Sciences, 34(1), 1466-1471.
21. Ge, Y. F., Orlowska, M., Cao, J., Wang, H., & Zhang, Y. (2022). MDDE: multitasking distributed differential evolution for privacy-preserving database fragmentation. The VLDB Journal, 1-19.
22. Castro-Medina, F., Rodríguez-Mazahua, L., López-Chau, A., Cervantes, J., Alor-Hernández, G., Machorro-Cano, I., & Arrioja-Rodríguez, M. L. (2022). A New Method of Dynamic Horizontal Fragmentation for Multimedia Databases Contemplating Content-Based Queries. Electronics, 11(2), 288.
23. Biyanto, T. R., Irawan, S., Febrianto, H. Y., Afdanny, N., Rahman, A. H., Gunawan, K. S., & Bethiana, T. N. (2017). Killer whale algorithm: an algorithm inspired by the life of killer whale. Procedia computer science, 124, 151-157.
24. Nagesh, H. S., Goil, S., & Choudhary, A. (2000). A scalable parallel subspace clustering algorithm for massive data sets. In Proceedings 2000 international conference on parallel processing (pp. 477-484). IEEE.
25. Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. IEEE transactions on pattern analysis and machine intelligence, 35(11), 2765-2781.