

# AN IMPROVED VOGEL'S APPROXIMATION METHOD (IVAM) FOR FRAGMENT ALLOCATION AND REPLICATION IN DISTRIBUTED DATABASE SYSTEMS

Mukta Aggarwal

Research Scholar, ASET, Amity University Haryana, Gurugram  
Gurugram, Haryana, 122413, India  
rishu.muk@gmail.com

Shalini Bhaskar Bajaj

Professor, ASET, Amity University Haryana, Gurugram  
Gurugram, Haryana, 122413, India  
shalinivimal@gmail.com

Vivek Jaglan

Professor, Computer Science, DPG Institute of Technology and Management, Gurugram  
Gurugram, Haryana, 122004, India  
jaglanvivek@gmail.com

## Abstract

In distributed database, allocation of fragments depends on the access time, response time, transmission cost and storage capacity. An Improved VAM mathematical model is introduced to allocate and replicate fragments in an optimal way either by maximizing the access frequency or minimizing the transmission cost. The method reduces the migration of fragments over the network by allocating fragments to the sites where the access frequency is higher. Storage capacity of sites is another essential constraint considered here. Successful retrieval ratio of fragments is taken as a performance measure to show the efficacy and the optimal behavior of the proposed algorithm. It worked on 3 fragments and 4 sites and obtained SRR is .26 for allocation and .24 for replication which is higher than compared works. For 3 fragments and 6 sites, SRR for allocation is .191 and for replication is .17 which has optimally increased the successful retrieval ratio.

**Keywords:** Access Frequency Counter (AFC); Fragment Size (FS); Storage Capacity (SC); Successful Access Frequency (SAF); Successful Retrieval Ratio (SRR); Total Access Frequency (TAF).

## 1. Introduction

Database has become an essential part for keeping records in all organizations. A centralized database has all its data at central place and distributed database has distributed data at various sites. Due to the increasing demand of data by many applications and disadvantages of centralized systems, distributed databases have gained importance. Development of distributed databases helps in increasing the reliability and availability of data and in decreasing the access time of data from various sites. Distributed databases can be expanded easily. Distribution of data can be done at table level. Any given table may be partitioned and stored at different sites and this technique is known as Fragmentation and Allocation.

Fragmentation can be done using various methods: horizontal fragmentation [Bhuvar *et al.* (2012)] and vertical fragmentation [Goel and Bajaj (2018)]. These fragments contain sufficient information to allow reconstruction of the original table [George and Balakrishnan (2015)]. The problem of fragmenting a table is a difficult one and many approaches exist for fragmenting databases [Elmasri and Navathe (2016)], [Silberschatz *et al.* (2014)]. This paper assumes that the database has already been fragmented. After fragmenting the database, allocation techniques are required to distribute fragments among various sites. Fragmentation is tightly coupled with allocation as in a distributed database system, fragments are formed to keep the data closer to the user who needs it frequently and this can be done using an appropriate allocation scheme. Proper allocation reduces the

access time of the fragment. Performance of system may improve by reducing response time and maximizing the throughput.

Fragment allocation can be done either by allocating exactly one copy of each fragment i.e., non-redundant allocation, or allocating replicas of fragments at more than one site i.e., redundant allocation. Redundant allocation i.e., replication, can be full or partial. In full replication, a copy of fragment is allocated at every site and in partial replication copy of fragment is stored at few sites.

Data allocation and replication problem can be handled more efficiently by using suitable mathematical models or optimization principles. In this paper, an Improved Vogel's Approximation Method (IVAM) mathematical model is used for allocation and replication of fragments.

Main concern of a distributed Database System is to fragment the underlying database and allocate them to the sites. A lot of work has been done on fragmentation and allocation from time to time, but still there is vast scope of efficiently allocating fragments to sites. [Khan and Hoque (2010)] presented the horizontal fragmentation and allocation scheme. Allocation is done based on predicate value. In [Raouf *et al.* (2015)], vertical fragmentation, allocation as well as replication are considered. Fragments are formed based on simultaneous access of attributes by various sites. For allocation, only attribute manipulate (update) operations for each fragment are calculated at each site. The fragment having maximum number of manipulate operations on a site is allocated to that site. For replication only attribute read operations are calculated for each fragment at each site and the fragment is replicated at the site at which maximum read operations occurred. If the site is same as in allocation step, then replica of that fragment is assigned to next higher value of read operations of the fragment. In [Huang and Chen (2001)] a simple and comprehensive model for allocation is discussed that reflects the transaction behavior in distributed databases. Two heuristic algorithms are developed to find the near optimal solution. In [Abdalla and Amer (2012)], authors discussed the synchronized fragmentation and allocation of a distributed database. Communication cost between sites and site capacity as a constraint are considered for allocation.

In [Abdalla (2014)], a new method to satisfy a synchronized horizontal fragmentation and allocation using a novel cost model is proposed. Fragment Allocation is based on the access pattern of fragments and the cost that is calculated due to the movement of data fragments from one site to another (data transmission cost). The total cost is the sum of retrieval cost, update, and storage cost. This total cost function is optimized for fragment allocation and the optimized result provides allocation that leads to fragment distribution. This approach minimizes frequent remote access and results in reducing data transfer cost. In [Mukherjee (2011)], initial allocation is applied statically. After the initial allocation, an access counter matrix at each site is maintained by the system to keep track of every fragment that is stored on the same site. If an access is made to any fragment, then access counter of the fragment at accessing site is increased by one. The node with highest access counter value is called owner node. If counter of remote site become greater than the counter of current owner, then the fragment is migrated to the accessing node and the node becomes new owner. The major drawback is that if the access pattern for each fragment changes frequently then it will take more time in transferring fragments and will increase data transmission rate. The migration of fragments is decreased in [Ulus and Uysal (2003)], by applying the constraint on the stay of fragments. After a migration to the new node, the stay of fragment must be there for a threshold value.

In [George *et al.* (2012)], fragment reallocation is optimized using a mathematical model, Hungarian assignment algorithm. Initially fragments are allocated based on assumption. If the query initiated on a site do not find fragment, then it is forwarded as an un-successful query. This leads to increase in the time taken in serving an unsuccessful query. While allocating fragments in distributed database, storage capacity of sites and fragment size is also important which is not considered in this algorithm.

In [Karimiadl *et al.* (2009)], the total transaction response time under memory capacity constraints of the sites is minimized using ant colony optimization algorithm. [Ahmad and Karlapalem (2002)] addressed the problem of non-redundant data allocation of fragments in distributed database system. A query execution strategy integrated with a query driven data allocation approach is developed. A site-independent fragment dependency graph of data transfer costs is developed which is incurred in executing a query. A uniform framework is developed to define and solve data allocation problems. In [Abdalla *et al.* (2014)], [Amer *et al.* (2017)], authors combined the horizontal fragmentation and allocation techniques. During allocation transmission cost was reduced and data locality was increased. Highest access cost parameter is used to place the fragments in the cluster. The average update cost (AUC) and average retrieval cost (ARC) were calculated for each fragment and update and retrieval queries were applied based on these parameters. Whenever fragment's  $AUC > ARC$ , the fragment is to be reassigned to the site based on highest update cost. While applying the process, there should be no violation of site constraints else the fragment will be placed at the site with next highest AUC. In [Abdalla (2012)], a two-phase allocation method is used with vertical fragmentation to minimize the communication cost and query response time. The updated cost values for each individual fragment are used. For improving the distributed system performance, communication cost is minimized in allocation and replication [Abdalla and Artoli (2019)]. In [Kumar and Gunasekaran (2020)], replication is used in enhancing the performance in e-learning.

It is observed from above that several researchers have submitted data allocation and replication techniques which attempt to minimize the total cost [Gopinath and Sherly (2018)]. The total cost (TOC) has two components: storage cost (STC) and query processing cost (QPC) [Ozsu and Valduriez (2011)].

$$\text{TOC} = \text{STC} + \text{QPC} \quad (1)$$

Storage cost depends upon the size of the fragment. Query processing cost consists of two components- processing cost and transmission cost. Processing cost (PC) is defined as the time taken to execute a query at a site. Transmission cost (TC) is defined as the cost to transfer data from one site to another.

$$\text{QPC} = \text{PC} + \text{TC} \quad (2)$$

According to literature, for appropriate allocation, either transmission cost should be less, or access frequency of the fragments should be high. Earlier works has either discussed about access frequency or transmission cost, but both these problems are not been solved by researchers simultaneously. Vogel's Approximation Method (VAM) is a mathematical model that is frequently used to solve optimization problems such as demand and supply problems, job assignment problems etc. either by maximizing the profit or minimizing the cost [Gupta and Hira (2014)]. But VAM has not been used for allocating fragments in distributed environment till date. Moreover, the VAM algorithm is assigning the units by dividing them according to the capacity available. If the classic VAM is applied on fragment allocation, then fragments will further decimate into smaller fragments, which is not preferable.

This paper proposes IVAM mathematical model for allocation and replication, which allocates the whole fragment at the site where the access frequency is high, and enough storage capacity is available. If storage capacity is not sufficient then fragment is allocated at some another site. Also, the second highest frequency is taken into consideration for replicating the fragments at other sites which is not done in VAM method. IVAM is considering the storage constraint and simultaneously maximizing the total access frequency. Another factor that is of prime significance is the transmission cost. This method can also be used for allocation in such a way that the transmission cost can be reduced. IVAM works in both situations either by maximizing the access frequency or by minimizing transmission cost.

## 2. Improved Vogel's Approximation Method (IVAM)

In distributed databases, finding an optimal or even a good solution for data allocation is a complex problem. IVAM method is an iterative procedure for computing a feasible and optimized solution of the allocation problem.

### 2.1. Problem description

Assume that there is a set of fragments  $F_i = \{F_1, F_2, \dots, F_m\}$ , where  $i = 1, 2, \dots, m$ , and a distributed system consisting of sites  $S_j = \{S_1, S_2, \dots, S_n\}$ , where  $j = 1, 2, \dots, n$ , on which a set of applications is running. The proposed algorithm finds the optimal distribution of  $F_i$  to  $S_j$  by maximizing the access frequency count. Also, each Site has a fixed Storage Capacity ( $SC_j$ ) and each Fragment Size ( $FS_i$ ) is fixed and this size can be calculated by the data type of attributes involved in a fragment.

If  $i^{\text{th}}$  fragment is accessed by  $j^{\text{th}}$  site, then it is represented as 1, otherwise 0. An access frequency counter ( $AFC_{ij}$ ) is used to keep the record of number of accesses of a fragment  $F_i$ , at site  $S_j$ . AFC value is increased by 1 if  $F_i$  is accessed again at  $S_j$ .

Consider that there is a set of fragments  $F_i = (F_1, F_2, F_3, F_4)$  where  $m=4$ , and a distributed system consisting of sites  $S_j = (S_1, S_2, S_3, S_4)$  where  $n=4$ , on which a set of applications is running. In Table 1, Fragments are represented as columns, sites are represented as rows and each cell shows  $AFC_{ij}$  value of fragment  $F_i$ , at site  $S_j$ .  $SC_j = (50, 50, 60, 60)$  shows the storage capacity of site  $S_j$  and  $FS_i = (30, 40, 50, 40)$  represents the size of fragment  $F_i$ .

Table 1 represents the  $AFC_{ij}$  values and if the aim is to reduce the transmission cost, then these  $AFC_{ij}$  values will be taken as Transmission Cost ( $TC_{ij}$ ).

## 3. Methodology

### 3.1. Proposed algorithm

Allocation of  $F_i$  is done at site  $S_j$  where the  $AFC_{ij}$  value is maximum. If allocation is done at site having maximum  $AFC_{ij}$  value, then transmission of fragments between sites is reduced. Simultaneously a check is maintained over the capacity of the site. It should be enough to store the fragment and satisfy the following condition:

$$\sum_{F_i \in F} FS \text{ of } F_i \leq SC \text{ of } S_j \quad (3)$$

The following IVAM algorithm steps are followed to allocate fragments at the appropriate sites:

Step 1: Check the problem: a.) to maximize the AFC value, go to step 2.

- b.) to minimize transmission cost, go to step 4.
- Step 2: Multiply the matrix by -1 to convert the maximization problem into minimization problem.
- Step 3: Add the highest AFC value to each cell, to make all the values positive.
- Step 4: Find the cells having smallest and next to smallest value in each row and write the difference as Row Difference (RD) alongside of the table in each row.
- Step 5: Find the cells having smallest and next to smallest in each column and write the difference as Column Difference (CD) under each column.
- Step 6: Select the highest value out of RD and CD and select the corresponding row or column and find the cell that has least value. Allocate the fragment to the site considering the storage constraint of that row or column and deduct the size of fragment from the site capacity. If there is a tie in the values of RD and CD, then select the cell for allocation based on the storage capacity.
- Step 7: Adjust the site Storage Capacity (SC) and Fragment Size (FS) and strike out the allocated row or column.
- Step 8: Repeat step 4 to step 7 until all fragments are allocated.

### 3.2. Successful Retrieval Ratio (SRR)

After applying the algorithm for maximizing the AFC value, Successful Retrieval Ratio is used as a performance measure to show the efficacy of the proposed method. If a request is made for a fragment and it is found at local site, then it is called successful retrieval and the access time and response time are said to be reduced.

SRR is the fraction of Successful Access Frequency (SAF) to the Total Access Frequency (TAF). It is calculated as:

$$SRR = SAF / TAF$$

Where SAF is the number of attempts made to the fragment by any application running on the same site and is calculated as in Eq. (4):

$$SAF = \sum_{i=1}^m \sum_{j=1}^n AFC_{ij} \quad \text{where } F_i \text{ is allocated to } S_j \quad (4)$$

Total Access Frequency (TAF) is the sum of the access frequency of each fragment on each site and is calculated as in Eq. (5):

$$TAF = \sum_{i=1}^m \sum_{j=1}^n AFC_{ij} \quad (5)$$

### 3.3. Allocation using IVAM to maximize AFC

Here, to minimize the transmission cost, IVAM algorithm starts from step 4 and to maximize the AFC value, algorithm starts from step 2. The following section applies the algorithm on maximizing the AFC value taking into consideration the storage constraint. Table 2 is obtained by applying step 2 and 3, where each AFC value is multiplied by -1 and highest AFC value, i.e., 18 is added to all AFC values.

Using step 4, for each row in Table 2, RD is calculated and tabulated in Table 3. For example, in first row two least values are  $AFC_{11}=8$  and  $AFC_{31}=12$  and their difference is 4. By applying step 5 at each column in Table 2, CD is calculated and tabulated in Table 3. For example, in first column two least values are  $AFC_{11}=8$  and  $AFC_{13}=0$  and their difference is 8. From the current iteration, the highest value out of RD and CD is 10 (Table 3). RD is highest, so check all  $AFC_{i3}$  values of the row.  $AFC_{13}=0$  is minimum. Allocate  $F_1$  to  $S_3$  and FS ( $F_1$ ) will become 0, as shown in Table 5.  $F_1$  is strike out. Remaining capacity for  $S_3$  is  $60-30=30$  (Table 5).

Now again RD and CD are calculated and tabulated in Table 4. Here, maximum value out of RD and CD is 7. So,  $F_2$  is allocated to  $S_4$ . Now, FS of  $F_2$  is assigned as 0 and SC of  $S_4$  is updated to  $60-40=20$  (Table 5). Assign FS( $F_2$ ) = 0.

$S \downarrow / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)
$S_1$	10	5	6	3	50
$S_2$	6	5	8	7	50
$S_3$	18	8	2	7	60
$S_4$	10	15	12	6	60
Fragment Size (in bytes)	30	40	50	40	-

Table 1. Access Frequency of Fragments at each Site with Storage Capacity of sites and fragment size

$S_{\downarrow} / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)
$S_1$	8	13	12	15	50
$S_2$	12	13	10	11	50
$S_3$	0	10	16	11	60
$S_4$	8	3	6	12	60
Fragment Size (in bytes)	30	40	50	40	

Table 2. Multiply the Access Frequency of Fragments by -1 and adding Highest AFC value

$S_{\downarrow} / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)	Row Diff
$S_1$	8	13	12	15	50	4
$S_2$	12	13	10	11	50	1
$S_3$	0	10	16	11	60	10
$S_4$	8	3	6	12	60	3
Fragment Size (in bytes)	30	40	50	40		
Col Diff	8	7	4	0		

Table 3. Allocation of First fragment

$S_{\downarrow} / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)	Row Diff (1)	Row Diff (2)
$S_1$	8	13	12	15	50	4	1
$S_2$	12	13	10	11	50	1	1
$S_3$	0	10	16	11	30	10	1
$S_4$	8	3	6	12	60	3	3
Fragment Size (in bytes)	0	40	50	40	-	-	-
Col Diff (1)	8	7	4	0	-	-	-
Col Diff (2)	-	7	4	0	-	-	-

Table 4. Allocation of second fragment

$S_{\downarrow} / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)	Row Diff (1)	Row Diff (2)	Row Diff (3)
$S_1$	8	13	12	15	50	4	1	3
$S_2$	12	13	10	11	50	1	1	1
$S_3$	0	10	16	11	30	10	1	5
$S_4$	8	3	6	12	20	3	3	6
Fragment Size (in bytes)	0	0	50	40	-	-	-	-
Col Diff (1)	8	7	4	0	-	-	-	-
Col Diff (2)	-	7	4	0	-	-	-	-
Col Diff (3)	-	-	4	0	-	-	-	-

Table 5. Allocation of Third fragment

$S_{\downarrow} / F \rightarrow$	$F_1$	$F_2$	$F_3$	$F_4$	Storage Capacity (in bytes)
$S_1$	10	5	6	3	10
$S_2$	6	5	8	7	0
$S_3$	18	8	2	7	30
$S_4$	10	15	12	6	20
Fragment Size (in bytes)	0	0	0	0	

Table 6. Allocation of all Fragments at Sites where Access Frequency Count are High

In Table 5, following step 6, highest value is 6 and the corresponding  $AFC_{34} = 6$  is the least value. But size of fragment  $FS(F_3) > SC(S_4)$ , so allocation is not possible here due to storage constraint violation according to step 6. Next highest value is 5 and  $AFC_{43} = 11$  is the least value. This is also not possible because  $FS(F_4) > SC(S_3)$ . The next highest value is 4 and  $AFC_{34} = 6$  is minimum but still can't be allocated.  $AFC_{32} = 10$  is next highest value and as storage capacity is more than fragment size,  $F_3$  is allocated at site  $S_2$ . FS of  $F_3$  is assigned as 0 and SC of  $S_2$  is updated to  $50-50=0$  (Table 6). Strike out column  $F_3$ . This step shows the importance of storage capacity of sites. In earlier works, storage constraint is considered but it is not discussed that what measure should be taken if constraint is not satisfied. This approach clearly shows that how to pick next site if constraint is not satisfied and is performing meticulously.

Site	Fragment	Access Frequency Count
$S_1$	$F_4$	3
$S_2$	$F_3$	8
$S_3$	$F_1$	18
$S_4$	$F_2$	15

Table 7. Final Allocation after Applying IVAM

Only Fragment  $F_4$  is left and is allocated to site  $S_1$ . Table 6 shows the allocation of all fragments with highest AFC values and the final allocation is represented in Table 7.

### 3.4. Successful Retrieval Ratio using IVAM

As stated earlier, the performance of the proposed algorithm is calculated in terms of SRR. Table 7 shows that the AFC values obtained are 3, 8, 18 and 15. So, SAF is calculated as:

Successful Access Frequency (SAF) = 44 using eq. (4)

Total Access Frequency (TAF) = 128 using eq. (5)

Successful Retrieval Ratio =  $44/128 = 0.344$

Proposed algorithm allocates fragments to sites considering both AFC value and storage capacity constraint. To show the meticulous performance of IVAM algorithm, it is applied on the data of earlier research works and is compared and analyzed for both AFC maximization and then for transmission cost minimization.

### 3.5. Replication

Replication of fragment is a process of storing copies of fragments at two or more sites. Replication is required in case more than one sites have frequent access to same fragment. The advantages of replication are reliability, availability, reduction in network load and parallelism. In the proposed work, IVAM method is applied for replication also. Fragments are replicated at the sites where the AFC value is second highest, and the fragment is already not allocated to that site. The results of replication are compared and shown in Results and Discussion.

## 4. Results and Discussion

### 4.1. For maximizing Access Frequency Count

For comparison, initial data of AFC is taken from [Abdalla *et al.* (2014)], [Amer *et al.* (2017)]. It consists of three fragments and four sites. Size of fragment is calculated by adding the attributes length in bytes. Access Frequency Count (after addition of query retrieval and update frequency of the attributes) is shown in Table 8. Table 9 and Table 10 gives the final allocation and replication of fragments to sites. Table 11 gives the final allocation and replication using the proposed IVAM algorithm on the same set of data.

Table 12 and 13 give the comparison of Successful Access Frequency and Successful Retrieval Ratio of the proposed work for  $m=3$  and  $n=4$ . Fig. 1 and Fig. 2 show the comparison of SAF and SRR for the same. As shown, the SAF of IVAM is 2190 for allocation and 1965 for replication. In both cases, either it is equal or greater than the compared works. Hence, the objective of achieving high SRR is obtained due to optimal allocation by mathematical model.

Table 14 gives Access Frequency Count (after addition of query retrieval and update frequency of the attributes) for three fragments and six sites [Abdalla *et al.* (2014)], [Amer *et al.* (2017)]. Table 15 and Table 16 shows the corresponding final allocation and replication of fragments to sites. Table 17 shows the final allocation and replication for three fragments and six sites using proposed algorithm.

Table 18 and 19 give the comparison of Successful Access Frequency and Successful Retrieval Ratio of the proposed work for  $m=3$  and  $n=6$ . Fig. 3 and Fig. 4 show the comparison of SAF and SRR for the same. The graph

shows that by increasing the number of sites the scope of allocating fragments is increased and the results obtained using IVAM far outweighs the compared works.

$S \downarrow / F \rightarrow$	$F_1$	$F_2$	$F_3$	Storage Capacity (in bytes)
$S_1$	732	980	406	1000
$S_2$	546	740	423	900
$S_3$	882	1126	263	250
$S_4$	784	970	426	870
Fragment Size (in bytes)	270	360	180	

Table 8. Access Frequency Count (Fragments=3, Sites=4)

Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
$S_1$	$F_2$	980	$S_1$	$F_1$	732
$S_4$	$F_1$	784	$S_2$	$F_2$	740
$S_4$	$F_3$	426	$S_1$	$F_3$	406

Table 9. Allocation and replication Proposed by (Abdalla et al., 2014)

Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
$S_1$	$F_1$	732	$S_4$	$F_1$	784
$S_1$	$F_2$	980	$S_2$	$F_2$	740
$S_2$	$F_3$	423	$S_1$	$F_3$	406

Table 10. Allocation and replication Proposed by (Amer et al., 2017)

Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
$S_1$	$F_2$	980	$S_1$	$F_1$	732
$S_4$	$F_1$	784	$S_3$	$F_3$	263
$S_4$	$F_3$	426	$S_4$	$F_2$	970

Table 11. Allocation and replication using IVAM

	Improved VAM	Proposed by (Abdalla et al., 2014)	Proposed by (Amer et al., 2017)
Allocation	2190	2190	2135
Replication	1965	1878	1930

Table 12. Comparison of Proposed Method with earlier works for Successful Access Frequency (Fragments=3, Sites=4)

	IVAM	Proposed by (Abdalla et al., 2014)	Proposed by (Amer et al., 2017)
Allocation	0.26	0.26	0.25
Replication	0.24	0.22	0.23

Table 13. Comparison of Proposed Method with earlier works for Successful Retrieval Ratio (Fragments=3, Sites=4)

$S \downarrow / F \rightarrow$	$F_1$	$F_2$	$F_3$	Storage Capacity (in bytes)
$S_1$	624	824	224	1000
$S_2$	650	754	310	900
$S_3$	560	676	160	250
$S_4$	530	570	158	870
$S_5$	588	752	232	950
$S_6$	602	714	254	710
Fragment Size (in bytes)	270	360	180	

Table 14. Access Frequency Count (Fragments=3, Sites=6)

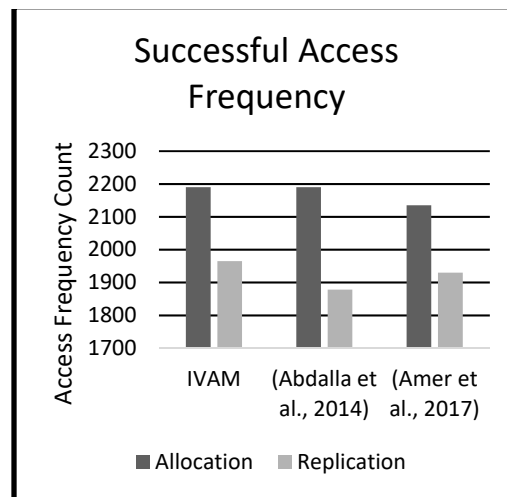


Figure 1. Comparison of Proposed method with earlier works for Fragments=3, Sites=4

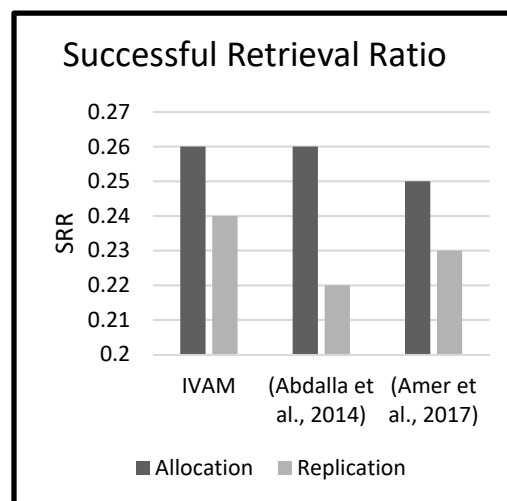


Figure 2. Comparison of Proposed method with earlier works for Fragments=3, Sites=4

Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
S <sub>3</sub>	F <sub>1</sub>	560	S <sub>2</sub>	F <sub>1</sub>	650
S <sub>7</sub>	F <sub>2</sub>	824	S <sub>4</sub>	F <sub>2</sub>	570
S <sub>1</sub>	F <sub>3</sub>	224	S <sub>3</sub>	F <sub>3</sub>	160

Table 15. Allocation and replication Proposed by (Abdalla et al., 2014)

Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
S <sub>2</sub>	F <sub>1</sub>	650	S <sub>4</sub>	F <sub>1</sub>	530
S <sub>6</sub>	F <sub>2</sub>	714	S <sub>1</sub>	F <sub>3</sub>	224
S <sub>6</sub>	F <sub>3</sub>	254	S <sub>4</sub>	F <sub>2</sub>	570

Table 16. Allocation and replication Proposed by (Amer et al., 2017)



Allocation			Replication		
Site	Fragment	Access Frequency Count	Site	Fragment	Access Frequency Count
S <sub>1</sub>	F <sub>2</sub>	824	S <sub>5</sub>	F <sub>1</sub>	588
S <sub>1</sub>	F <sub>1</sub>	624	S <sub>1</sub>	F <sub>3</sub>	224
S <sub>2</sub>	F <sub>3</sub>	310	S <sub>2</sub>	F <sub>2</sub>	754

Table 17. Allocation and replication using IVAM

	IVAM	Proposed by (Abdalla et al., 2014)	Proposed by (Amer et al., 2017)
Allocation	1758	1608	1618
Replication	1566	1380	1324

Table 18. Comparison of Proposed Method with earlier works for Successful Access Frequency (Fragments=3, Sites=6)

	IVAM	Proposed by (Abdalla et al., 2014)	Proposed by (Amer et al., 2017)
Allocation	0.191	0.175	0.176
Replication	0.17	0.15	0.14

Table 19. Comparison of Proposed Method with earlier works for Successful Retrieval Ratio (Fragments=3, Sites=6)

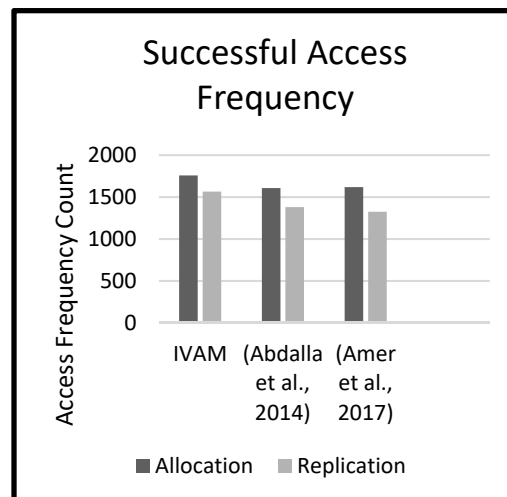


Figure 3. Comparison of Proposed method with earlier works for Fragments=3, Sites=6

For ensuring fair comparison same dataset as [Abdalla *et al.* (2014)], [Amer *et al.* (2017)] is incorporated for the allocation and replication. Table 12 and Table 13 give the comparison SAF and SRR respectively for both allocation and replication of proposed algorithm with previous works for  $m=3$  and  $n=4$ . Table 18 and Table 19 give comparison

of SAF and SRR respectively for both allocation and replication for  $m=3$  and  $n=6$ . As shown, in contrast to earlier works, the proposed algorithm has highest SRR for both the cases. The proposed algorithm is working optimally even in case of variable number of sites. The results shows that all fragments are allocated to the sites where AFC value for each fragment is highest or next to highest value depending upon the storage capacity of the site. IVAM algorithm provides an optimal allocation that leads to effective fragment distribution with less remote access by increasing the successful retrieval ratio.

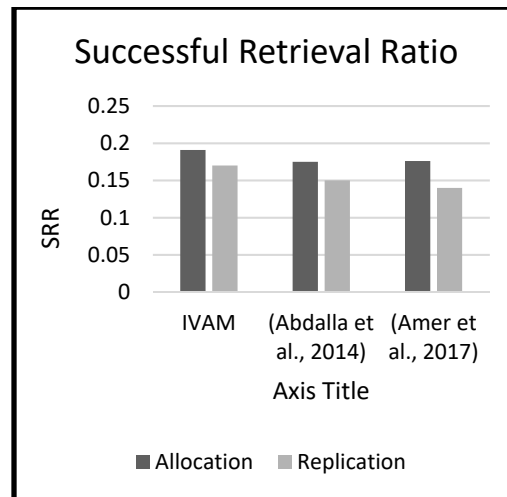


Figure 4. Comparison of Proposed method with earlier works for Fragments=3, Sites=6

#### 4.2. For minimizing transmission cost

Transmission cost is composed of the total frequency of each executed query, cost of communication between sites, size of the fragment, number of sites involved in responding the query, and if a particular fragment is allocated or not at a specific site [Amer *et al.* (2020)]. Using the proposed algorithm, transmission cost is minimized. For comparison, initial data of Transmission Cost [Amer *et al.* (2020)] is shown in Table 20. It consists of two fragments and six sites. Size of fragment is calculated by adding the attributes length in bytes. Storage capacity for each site is assumed to be equal. Transmission Cost in allocation and replication of each fragment at every site is shown in Table 21. Table 22 gives the final allocation and replication of fragments to sites using the proposed IVAM algorithm on the same set of data.

Table 21 and Table 22 show that the transmission cost of allocation is same for [Amer *et al.* (2020)] and IVAM i.e., 256190. The replica of both fragments ( $F_1$  and  $F_2$ ) in IVAM are occurring at the same site which leads to decrease in transmission cost. The storage constraint of sites is again considered while allocating the fragments to sites. The transmission cost of [Amer *et al.* (2020)] is 286790 and for IVAM it is 284990. This shows that the replication transmission cost of IVAM is reduced as compared to [Amer *et al.* (2020)].

In the proposed work, the allocation process is accomplished for fragments over network sites taking into consideration the storage constraint of sites by using mathematical model IVAM. IVAM method is an iterative procedure for computing a feasible and optimized solution of the allocation problem. If a request is made for a fragment and it is found at local site, then it is called successful retrieval and the access time and response time are said to be reduced. For allocation, storage constraint of each site is also important. So, to optimally control these parameters, fragments are allocated to sites where the access frequency of those fragments is maximum, and enough storage capacity is available. Replication is required in case more than one sites have frequent access to same fragment. In the proposed work, IVAM method is applied for replication also. Fragments are replicated at the sites where the AFC value is second highest, and the fragment is not allocated to that site.

$S \downarrow / F \rightarrow$	$F_1$	$F_2$	Storage Capacity (in bytes)
$S_1$	398090	41100	50
$S_2$	376940	41100	50
$S_3$	242990	15600	50
$S_4$	306440	35100	50
$S_5$	271190	13800	50
$S_6$	297510	13200	50
Fragment Size (in bytes)	46	4	

Table 20. Data set for Allocation by minimizing Transmission Cost (Fragments=2, Sites=6)

Allocation			Replication		
Site	Fragment	Transmission Cost	Site	Fragment	Transmission Cost
S <sub>4</sub>	F <sub>1</sub>	242990	S <sub>3</sub>	F <sub>1</sub>	271190
S <sub>5</sub>	F <sub>2</sub>	13200	S <sub>4</sub>	F <sub>2</sub>	15600

Table 21. Allocation and replication Proposed by (Amer et al., 2020)

Allocation			Replication		
Site	Fragment	Transmission Cost	Site	Fragment	Transmission Cost
S <sub>4</sub>	F <sub>1</sub>	242990	S <sub>3</sub>	F <sub>1</sub>	271190
S <sub>5</sub>	F <sub>2</sub>	13200	S <sub>3</sub>	F <sub>2</sub>	13800

Table 22. Allocation and replication using IVAM

## 5. Conclusion and Future scope

In distributed database, fragments are distributed among the sites connected within a network. This paper addresses the issue of fragment allocation and replication by increasing the successful queries using a mathematical model IVAM. Here, an example using IVAM is introduced with four fragments and four sites. Initially fragments are allocated on first come first serve basis. Weight is given to access frequency of fragments at each site and this information is maintained using an AFC. This improves the Successful retrieval ratio as compared to initial allocation and earlier works. Cases with varying number of sites are considered. The proposed algorithm works effectively for variable number of sites and fragments. This approach also covers the storage constraint aspect. Both maximum access frequency and storage capacity are evaluated simultaneously. Allocation of multiple fragments at a single site is also possible if the sufficient capacity to store fragments at the site is available. Replication is also done on the sites where the access frequency is high. This makes the process complete and reliable. Replication helps in improving the availability of data at sites but if the number of updates is large, it is advisable to limit the replication.

In future, this work may be extended considering the load balancing constraint and site clustering environment.

## 6. Conflicts of Interest

The authors declare no conflict of interest.

## References

- [1] Bhuyar, P. R., Gawande, A.D., and Deshmukh A. B., (2012). Horizontal Fragmentation Technique in Distributed Database. International Journal of Scientific and Research Publications, 2(5):7, 1.
- [2] Goel, M., and Bajaj, S. B., (2018). Comparative Analysis of Vertical Fragmentation Techniques in Distributed Environment. International Journal of Electrical Electronics & Computer Science Engineering, 5(1):5, 48.
- [3] George, D. I., and Balakrishnan, C., (2015). HAADAS: An Enhanced Approach for Re-Allocation of Fragments in Peer-to-Peer Distributed databases. International Journal of Applied Engineering Research, 10(82):6, 315.
- [4] Elmasri, R.; Navathe, S. B., (2016). Fundamentals of Database Systems. 7th ed. Pearson, NJ, 1242p.
- [5] Silberschatz, A.; Korth, H. F.; Sudarshan, S., (2014). Database System Concepts, 6th ed. McGraw Hill Education, NY, 1276p.
- [6] Khan S. I., and Hoque, A. S. M. L., (2010). A New Technique for Database Fragmentation. International Journal of Computer Applications, 5(9):5, 20.
- [7] Raouf, A. E. A., Badr, N. L., and Tolba, M. F., (2015). An Optimized Scheme for Vertical Fragmentation, Allocation and Replication of a Distributed Database. In 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS' 15), Dec 12- Dec 14, 2015; Cairo, Egypt, p. 506-513.
- [8] Huang, Y., and Chen, J., (2001). Fragment Allocation in Distributed Database Design. Journal of Information Science and Engineering, 17:16, 491.
- [9] Abdalla, H. I., and Amer, A. A., (2012). Dynamic Horizontal Fragmentation, Replication and Allocation Model in DDBSs. In IEEE International Conference on Information Technology and e-Services, Mar 24-Mar 26, 2012; Sousse, Tunisia, p. 1-6.
- [10] Abdalla, H. I., (2014). A Synchronized Design Technique for Efficient Data Distribution. Computers in Human Behavior (Elsevier), 30:9, 427.
- [11] Mukherjee, N., (2011). Synthesis of Non-Replicated Dynamic Fragment Allocation Algorithm in Distributed Database Systems. International Journal on Information Technology (ACEEE), 1(1):6, 36.
- [12] Ulus, T., and Uysal, M., (2003). Heuristic Approach to Dynamic Data Allocation in Distributed Database Systems. Pakistan Journal of Information and Technology, 2(3):9, 231.
- [13] George, D. I., Balakrishnan, C., and Charles, A., (2012). An Improved Methodology for Fragment Re-allocation in Peer-to-Peer Distributed Databases, In 4th International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2012) (IEEE Explore), Oct 19-Oct 20, 2012, Bangalore, India, p. 78-81.
- [14] KarimiAdl, R., Taghi, S. M., and Rankoohi, R., (2009). A New Ant-Colony Optimization Based Algorithm for Data Allocation Problem in Distributed Database. Knowledge and Information System, 20(3): 25, 349.

- [15] Ahmad, I., and Karlapalem, K., (2002). Evolutionary Algorithms for Allocating Data in Distributed Systems. *Distributed and Parallel Databases*, 11:38, 5.
- [16] Abdalla, H. I., Amer, A. A., and Mathkour, H., (2014). Performance optimality enhancement algorithm in DDBS (POEA, *Computers in Human Behavior* (Elsevier), 30:8, 419.
- [17] Amer, A. A., Sewisy, A. A., and Elgendy, T. M. A., (2017). An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). *Heliyon* (Elsevier), 3(12):42, 1.
- [18] Abdalla, H. I., (2012). A New Data Re-Allocation Model for Distributed Database System. *International Journal of Database Theory and Application*, 5(2):16, 45.
- [19] Abdalla, H., and Artoli, A. M., (2019). Towards an Efficient Data Fragmentation, Allocation and Clustering Approach in a Distributed Environment. *MDPI, Information*, 10(3):20, 1.
- [20] Kumar, P. S., and Gunasekaran, M., (2020). An improved Vertical Fragmentation, Allocation and Replication for enhancing e-learning in Distributed Database Environment. *Computational Intelligence*, 37(1):20, 253.
- [21] Gopinath, S., and Sherly, E., (2018). A Comprehensive Survey on Data Replication techniques in Cloud Storage Systems. *International Journal of Applied Engineering Research*, 13(22):7, 15926.
- [22] Ozsu, M. T.; Valduriez, P., (2011). *Principles of Distributed Database Systems*. 3rd ed. Springer, NY, 846p.
- [23] Gupta, P. K.; Hira, D. S., (2014). *Operations Research*. 7th ed, S. Chand, India, 1512p.
- [24] Amer, A. A., Mohamed, M. H., and Al\_Asri, K., (2020). ASGOP: An Aggregated similarity-based greedy-oriented approach for relational DDBSs design. *Computer Science. Heliyon, Elsevier*, 6(1):17, 1.

## Authors Profile



**Mukta Aggarwal**, pursuing Ph.D. from Amity University Haryana, Gurugram. Currently working as Assistant Professor in The Technological Institute of Textile and Sciences, Bhiwani, Haryana. M.E. in CSE from NITTR, Chandigarh (Panjab University). B.Tech. from JMIT, Radaur (Kurukshetra University). She has teaching experience of 18 years. She has published 19 papers in various journals and conferences. She has supervised 6 M. Tech. students. Her research interests include image processing, image steganography, association rule mining and distributed databases.



**Dr. Shalini Bhaskar Bajaj** is Director of Amity School of Engineering and Technology and working as Professor and Head, Computer Science and Engineering, Amity University Haryana. She received her B.E. degree in Computer Science and Engineering from Deenbandhu Chhotu Ram University (formerly C.R.S.C.E., Murthal), Haryana, India, M.E. in Computer Technology and Applications from Delhi Technological University (formerly Delhi College of Engineering), Delhi, India, and Ph.D. in Information Technology from IIT, Delhi, India. Her research interests include databases, social network analysis, classification, clustering, association rule mining, temporal mining, data stream mining, structured data mining to name a few. She has attended various national and international conferences in India and abroad. She has published more than 50 research papers in various national and international conferences and journals.



**Dr. Vivek Jaglan** is working as Director/Principal and Professor (Computer Science & Engineering) in DPG Institute of Technology and Management, India. He has combined research and teaching experience of over 16 years in the field of artificial intelligence and have made significant contributions towards AI's applications in network optimisation, network security and prediction models. He has supervised seven Ph.D students and eleven Masters students completion and currently supervising four Ph.D students. He has published over 70 high quality research publication in reputed international journals and conferences, majorly in the field of Artificial Intelligence and its applications. He has two design patents (India).