

# Efficient Heuristic Replication Techniques for High Data Availability in Cloud

H. L. Chandrakala<sup>1,\*</sup> and R. Loganathan<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, School of Engineering, Presidency University, India

<sup>2</sup>Department of Computer Science and Engineering, HKBK College of Engineering, India

\*Corresponding Author: H. L. Chandrakala. Email: chandra.729354@gmail.com, chandrakalahl@gmail.com

Received: 27 January 2022; Accepted: 10 April 2022

**Abstract:** Most social networks allow connections amongst many people based on shared interests. Social networks have to offer shared data like videos, photos with minimum latency to the group, which could be challenging as the storage cost has to be minimized and hence entire data replication is not a solution. The replication of data across a network of read-intensive can potentially lead to increased savings in cost and energy and reduce the end-user's response time. Though simple and adaptive replication strategies exist, the solution is non-deterministic; the replicas of the data need to be optimized to the data usability, performance, and stability of the application systems. To resolve the non-deterministic issue of replication, metaheuristics are applied. In this work, Harmony Search and Tabu Search algorithms are used optimizing the replication process. A novel Harmony-Tabu search is proposed for effective placement and replication of data. Experiments on large datasets show the effectiveness of the proposed technique. It is seen that the bandwidth saving for proposed harmony-Tabu replication performs better in the range of 3.57% to 18.18% for varying number of cloud data-centers when compared to simple replication, Tabu replication and Harmony replication algorithm.

**Keywords:** Cloud computing; data replication; bandwidth saving; Tabu search; Harmony search; hybrid Harmony- Tabu search

## 1 Introduction

Cloud computing has been defined as a deep revolution that provides services based on information technology using virtualization technology. It considers all the ideas of distributed and parallel computing for offering on-demand computing resources to either the computers or any of the other devices [1]. The aim is to provide shared services with a high level of scalability, dynamicity, and reliability. However, there are many different issues faced by this owing to the complex architecture on which it is based. In the case of the datacentres of the cloud, the issues of complexity and scalability may cause an increase in both cost and energy that are the two major challenges requiring attention and some careful study. Since the energy consumed and the carbon footprint of the infrastructures of the cloud are high, the investigators had attempted to identify some applicable methods for bringing down the consumption of



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

energy. In the recent few years, the primary focus of the investigators has been in bringing about a reduction in the cost, space, and energy.

The replication approach in cloud computing creates several copies of similar data and stores them in different sites. As a result, the replication of cloud data reduced the waiting time of users, improved the availability of data, and also increased the consumption of the bandwidth. There are two different classes to the replication techniques: the dynamic approach and the static approach. The dynamic replication creates and deletes replicas based on the factors like access pattern, bandwidth, storage and topology of the cloud. It reduces the bandwidth requirements, saves cost and storage space. The static replication method predetermines the both replica creation and placement. Though static methods are easy of implement, it cannot adapt to changes in bandwidth, user access and storage. Some of the popular replication algorithms are simple replication algorithm, adaptive replication algorithm.

In order to be able to tolerate the corruption of data, the technique of data replication has been adopted extensively in the cloud computing for providing a high data availability [2]. But the requirement of the QoS for the application has not been considered in the replication of data. During data corruption, the QoS requirement of this application cannot be supported constantly. The primary reason has been explained as below: Using various nodes in cloud computing, it becomes challenging to ask the nodes to achieve similar performance with varying capacity of the disks, memory, and their CPUs. For instance, Amazon EC2 has been a cloud platform that is realistic and heterogeneous [3]. This system of cloud computing has certain heterogeneous traits in the nodes.

**Motivation:** Owing to the heterogeneity of the node, data with a high-QoS application needs to be replicated in a node of low-performance (a node that has latencies of disk access and slow communication). If in case there is some data corruption in this node that runs a high-QoS application, this application is retrieved from a node with low performance. As this node of low performance also has a slow communication with latencies of disk access, a QoS requirement of a high-QoS application can further be violated. Thus, the replicas of the data need to be optimized to the data usability, performance, and stability of the application systems. The selection and placement of data replicas in the cloud while maintaining the quality of service is a challenging process and metaheuristic algorithms are used to optimize the data replication.

The TABU Search is a method of meta-heuristic optimization that was formalized for the combinatorial problems [4]. It is based on a search procedure of the neighborhood. The TABU Search further supervises the solution space exploration by considering a new set of completely diversified moves which are the neighbors in its current state. It further categorizes some moves to be the “TABU” that was also called forbidden moves to store them in a new array list called the TABU list. A format made to search the neighborhood for a certain solution that is different from one another. The technique also avoids cycling, which refers to preventing the execution of similar series of moves and conducting a search in the unexplored regions. The adaptive memory was used and this helped in monitoring local information and with some information in relation to the process of exploration. The main advantages of a TABU Search are: a TABU list may be applied to the discrete and the continuous spaces that guide a process of local search for exploring a solution space beyond local optimality for the more challenging problems. For this, the TABU Search obtains several solutions that rival and surpass all of the good solutions found earlier.

The Harmony Search (HS) is another new metaheuristic developed by Geem and inspired by the process of musical performance where a musician searches for the state of harmony. An improvisation to a musical instrument will correspond to a new decision variable found in the optimization and it has a range of pitch that corresponds to the range of a value. This algorithm has been capturing a lot of attention and has been applied successfully to solve a range of optimization problems. It effectively identifies the regions of high performance in a solution space inside a reasonable time frame. A distance bandwidth (BW) has been

adjusted dynamically to favour an exploration in its early stages and the exploitation in its final stages of the search process.

Data replication in the cloud has been addressed by employing a hybrid optimized replication using the TABU Search and HS algorithm. The literature related to this work is discussed in Section 2. Section 3 gives details on the techniques used here. Section 4 elaborates the results obtained along with its analysis and Section 5 concluded this work.

## 2 Literature Survey

The primary target of the survey conducted by Amutha et al. [5], has been a mechanized evaluation or the explanation of the proceeding events in the cloud data and its replication system. There was a total of 34 papers used to determine the standard publication which was IEEE. Each paper consists of data replication techniques and is classified into four groups: various kinds of systems of replication, the Survey on the replications system, the Data placement on the data replications systems, and finally the Fault tolerance on the data replications systems. An analysis of the accomplishment of such promulgated tasks was calculated in an effective manner based on an analysis metrics of the output, the time taken for accomplishment, the storage usage, the replica number, the recovery time, and the cost of replication. The work also assisted in the enhancement of the novel technologies that were used by means of the system of data replication.

To provide cost-effective availability, and also to minimize the time taken for response by the applications, and finally, for ensuring load balancing in cloud storage, a new replica method has been proposed by Mansouri et al. [6]. The placement of the replica is based on five different parameters which are mean service time, failure probability, storage usage, latency, and load variance. The replication should, however, be used wisely since the size of storage for every such site is limited. So, the site will have to limit itself to the important replicas alone. A new strategy for the replacement of replica was presented on the basis of the file and its availability, the last time there was a request made for a replica, the actual numbers of access and the actual size of the replica. The proposed algorithm was evaluated by using the CloudSim simulator and it offered a better performance compared to that of the other algorithms with regard to the mean response time, the effective usage of the network, the storage usage, frequency of replication, and load balancing.

Effective management of data like the availability and its efficient access is a critical need for applications. This may be further achieved by means of employing a data replication that can bring down the data access latency that is reduced and higher availability of data along with an improved system of load balancing. Furthermore, different applications need different quality-of-service (QoS). For supporting the needs of the QoS, a highly distributed technique of QoS-aware replication was proposed by Shorfuzzaman [7] for computing the locations of the optimal datacentres. Also, the strategy of replication primarily aims at the maximization of the satisfaction of the QoS for improving the availability of data and also for reducing latency access. This problem has been formulated by making use of dynamic programming. Lastly, the experiments of the simulation were performed by making use of a data access pattern for demonstrating the proposed technique's effectiveness.

Lin et al. [8] had proposed two different QoS-aware data replication (QADR) algorithms for the cloud. The first one adopted an idea that was intuitive with a high-QoS first-replication (HQFR) for performing replication of data. But this greedy algorithm is not able to minimize the cost of data replication with the number of data replicas that are QoS violated. For achieving the two minimum objectives, the next algorithm can transform the problem of the QADR into some well-known minimum-cost maximum-flow (MCMF) based problem. The application of the currently existing MCMF algorithm for solving the problem of the QADR and the next one produces an optimal solution for this problem of the QADR in

the time taken for the polynomial. Furthermore, there were some techniques of node combination for reducing the replication of data time. In simulations, the proposed algorithm's effectiveness was seen in the replication and the recovery of data.

The replication of objects in various sites is probably a solution for decreasing network traffic. A decision on what has to be replicated and what constraint optimization problem needs to be solved is decided in advance. The Genetic Algorithm (GA) based algorithm was proposed for solving problems when the read or write demands continue to be static and proved experimentally a superior quality of solutions that were obtained in comparison to the intuitive greedy method. But the static approach has a high runtime and is not quite useful in this case. A hybrid GA was introduced by Loukopoulos et al. [9] which includes the current distribution of replica and also computed another new one that makes use of the knowledge regarding the attributes to the network and the changes that occurred. Considering the pragmatic scenarios in the environments of distributed information, these algorithms are evaluated with regard to the capacity of storage for each site and the variations to the popularity of the objects and further examine a trade-off between its running time and the quality of the solution.

Ebadi et al. [10] presented a hybrid metaheuristic algorithm to resolve the NP-hard problem of determining the number of replicas required and their placement in cloud. The authors proposed to use Particle Swarm Optimization (PSO) for its global search capacity and TS for its local search capacity to find optimal replication for cloud. The experimental results demonstrated that the proposed hybrid PSO-TS replication method achieves lower cost and saves energy compared to existing replication methods.

Awad et al. [11] presented multi-objective (MO) metaheuristic algorithms based on PSO and ACO for addressing the replication in cloud. The proposed MO-PSO selected the replicas based on the most frequently accessed ones. The proposed MO-ACO is used for placement of the replicas, it depends on the shortest distance and availability of replicas. The simulation results demonstrated the effectiveness of the proposed methods compared against existing replication algorithms [12].

As observed from the literature survey, metaheuristic algorithms are commonly used to achieve optimized replication to save bandwidth, cost, and storage. The PSO, ACO, GA are popularly used. In this work, the Harmony Search (HS) is used as it has advantages like quick convergence, easy to implement, robust, scalable and less parameters. The main disadvantage is it may suffer from premature convergence. To mitigate this, it is hybridized with Tabu Search (TS) in this work.

### 3 Methodology

The section explains the simple and adaptive replication algorithm, the harmonized search algorithm, the TABU search, the proposed TABU search, and the Harmony Search algorithm.

#### 3.1 Simple Replication Algorithm (SRA)

A Simple Replication Algorithm (SRA) is a method of replication accepting that there are no replicas and the read or write occurrences that were identified and tend to persist in nature.

Taken on behalf of the site  $S^{(i)}$  and the object  $O_k$  stated that the *replication benefit* based value  $B_k^{(i)}$ , as per Eq. (1):

$$B_k^{(i)} = \frac{R_k^{(i)} - \left( \sum_{x=1}^M w_k^{(x)} O_k C(i, SP_k) - W_k^{(i)} \right)}{O_k} \quad (1)$$

The benefit expected in the Network Transfer Cost (NTC) and its terms have been stated, if the replication  $O_k$  has been completed at the  $S^{(i)}$ . The value was calculated utilizing a variance among that of the NTC that happened from all the existing read requests and an NTC that rises owing to the updates to

a replica that is repaid to the size of the object. There are some negative values of the  $B_k^{(i)}$   $k$  means which duplicates the  $k$ th object, and this is ineffective from its “local view” of that of the  $i$ th site. This means that the local NTC which is detected from that of the  $i$ th site is enlarged.

There is a list  $L^{(i)}$  that has been kept for the  $S^{(i)}$  that comprises all objects that can be replicated. There is also an  $O_k$  which can further be replicated at the  $S^{(i)}$  only in case the residual capacity of storage  $b^{(i)}$  of this site is larger compared to its size and also its benefit value which is positive. There is a list  $LS$  that has been kept by means of considering the sites with the “opportunity” of replicating objects. Further, another site  $S^{(i)} \in LS$  if the  $L^{(i)} \neq \emptyset$ . For each stage, there is a site  $S^{(i)}$  that is selected from the  $LS$  found in a fashion called round-robin and their benefit values fit to the  $L^{(i)}$  that are calculated. The primary benefit has been replicated and also the lists  $LS$ , the  $L^{(i)}$  is composed with a new parallel and nearest site value  $SN_k^{(i)}$ , which are consequently reorganized. There is an SRA algorithm that has been presented as per Fig. 1.

```

Initialize LS and all  $L^{(i)}$ 
WHILE  $LS \neq \emptyset$  DO
    BMAX=0, OMAX=NULL /*BMAX holds the current max  $B_k^{(i)}$  value
                        OMAX holds the identity of the object for which  $B_k^{(i)} = *BMAX*$  /
    Pick up a site  $S^{(i)} \in LS$  in a Round-Robin way
    FOR each  $O_k \in L^{(i)}$  DO
        Compute  $B_k^{(i)}$ 
        IF  $BMAX \leq B_k^{(i)}$  THEN  $BMAX=B_k^{(i)}, OMAX = k$ 
        ELSEIF ( $B_k^{(i)} \leq 0$  OR  $b^{(i)} < O_k$ ) THEN  $L^{(i)} = L^{(i)} - \{O_k\}$ 
    Replicate  $O_{OMAX}$ 
     $L^{(i)} = L^{(i)} - \{O_{OMAX}\}$  /*Remove OMAX object from the list of potentials to be replicated*/
    FOR all sites in LS update the relevant  $SN_{OMAX}^{(i)}$  field. /*Update "nearest sites"*/
     $b^{(i)} = b^{(i)} - O_k$  /*New remaining capacity*/
    IF  $L^{(i)} = \emptyset$  THEN  $LS=LS - \{S^{(i)}\}$  /*Remove  $S^{(i)}$  if there are no other candidates (objects) to be replicated*/
ENDWHILE

```

**Figure 1:** Simple replication algorithm

### 3.2 Adaptive Data Replication Strategy (ADRS)

Adaptive data replication strategy (ADRS), contains effective placement of replica, and also the strategy for replacement of replica was identified and this had two parts. They are:

- To meet a high availability, a high requirement of efficiency, and a high fault tolerance, there is a need to dynamically adjust the sites for placing some newer replicas according to the current environments of the cloud. This random replacement of replica results in an access skew because some sites remain idle. It leads to a load imbalance. But, there is an appropriate placement of replicas on an ultra-large-scale, and this can improve response time by means of generating some newer replicas and further storing them to light-load sites.
- In case there is sufficient storage space in the chosen site, the replica is stored and if it does not, some files are deleted based on:
  - The file availability;
  - The time a replica was previously requested;
  - The number of access;
  - The size of the file of its replica.

There are various accesses and the last time this replica had been requested, there was an indication of a probability of it needing this replica once again. So it is critical to replace large-sized files as this can bring down the replacement. Let  $R$  and  $R'$  be the old and new replication methods and let  $D_R$  and  $D_{R'}$ , represent the total NTC created by  $R$  and  $R'$ , respectively. The  $X''$  is a matrix  $M \times N$  (0, 1) with elements  $X''_{ik}$  being 1 if  $O_k$  is replicated at  $S^{(i)}$  under the  $R'$  method else assigned 0. When executing  $R'$  some replicas are created and others deleted. Assuming that replica creation is achieved by transferring a copy of the object from the respective primary site and that replica deletion incurs no cost, the total NTC, denoted by for  $I_{RR'}$ , realizing  $R'$  is given by Eq. (2):

$$I_{RR'} = \sum_{i=1}^N \sum_{k=1}^M X''_{ik} (1 - X_{ik}) O_k C(i, SP_K) \quad (2)$$

where  $X_{ik}$  is the allocation variable for the replication method  $R$ . The total benefit, denoted by  $V_{RR'}$ , for moving from the  $R$  method to the  $R'$  is given by Eq. (3):

$$V_{RR'} = D_R - (D_{R'} + I_{RR'}) \quad (3)$$

The pseudocode of ADRS is shown in Fig. 2.

```

For every (the unavailable requested file which is fl in the local site)
// Replica Placement
Calculating of the Cost Function for each site;
Finding a site with the minimum Cost Function value to store the replica of the fl;
// Replica Replacement
In case (sufficient space for a replica of the fl in its best site)
    Storing of the replica;

Else
{
    Creating a list L of the files available in its best site;
    Sorting the list L by using the V value in an ascending order;
    When (L does not remain empty && has insufficient space for a replica of the fl)
        Delete the first file from the list L in its best site;
    Storing of the replica;
}
}

```

**Figure 2:** Pseudocode for adaptive data replication strategy

### 3.3 Tabu Search Algorithm

The Tabu Search (TS) is a method of neighbourhood search that employs the “intelligent” search along with a technique of flexible memory for avoiding getting trapped in the local minimum. This solves problems of combinatorial optimization and the memory structures used were divided into three different categories:

- a) **The Short Term:** with a list of its solutions that were considered with a potential one in a TABU list revisited when an expiration point is reached.

- b) **The Intermediate-Term:** the rules of intensification intended to result in a bias towards the other promising areas in the search space.
- c) **The Long Term:** the rules of diversification that drive this search to newer regions.

Every solution  $x \in X$  will have an associated neighbourhood  $N(x) \subset X$ , in which the  $X$  denotes a set of some feasible solutions. There is another solution  $x' \in N(x)$  that may be reached from the  $x$  iteratively by means of an operation known as a move until such time a stopping criterion is satisfied. In an iteration, the TS will move to the admissible neighbour in case it results in an objective function deteriorating. To avoid this type of cycling, some of the moves will be forbidden from getting re-instantiated for a certain period of time by using a structure of short-term memory known as the recency-based memory that has been implemented using the TABU-list. The chosen attributes occurring in the solutions that have been visited recently will be stored and then labelled as TABU-active for preventing these solutions from being revisited. This span of iterations at the time an attribute is maintained is called the TABU tenure. This TABU status for a certain solution gets overridden at the time an aspiration criterion is satisfied. Additionally, there is a critical consideration for the TABU search which was the determination of a suitable strategy of the candidate list for bringing down the examination of the elements of the  $N(x)$ , for achieving another very effective trade-off between the final quality of the  $x'$  and the actual effort expended in order to obtain it [13]. The algorithm steps for a TABU Search (TS) Algorithm is depicted in Fig. 3.

- **Step 1:** Choose the initial solution  $i$  in the  $S$ . Set the  $i^* = i$  and  $k=0$ .
- **Step 2:** Set the  $k=k+1$  and also generate subset  $V$  of a solution in  $N(i,k)$  to ensure one of the TABU conditions is violated or one condition of aspiration holds. 8
- **Step 3:** Choose the best  $j$  in the  $V$  and set  $i = j$ .
- **Step 4:** If the  $f(i) < f(i^*)$  then set the  $i^* = i$ .
- **Step 5:** Update the TABU and the conditions of aspiration.
- **Step 6:** In case the stopping condition is arrived at, stop. Else move to Step 2.

**Figure 3:** Algorithm for Tabu Search (TS) algorithm

### 3.4 Harmony Search Algorithm

The algorithm of a Harmony Search [14] is a very simple and one of the latest approaches taking inspiration from the playing of a simultaneous band to search for optimal and answerable to the issues of optimization. This means in other words, identification of optimal solutions to the complex problems and also the performance of playing music. Initially, the game offered this solution in the year 2001 – based on the logic of Metaheuristic approaches for ensuring harmony in playing such similar music.

On the basis of this Harmony Algorithm, this timing has two components: The Harmony memory, the Sound volume regulation Random selection, the Bandwidth (BW), and finally the level of Harmony memory. These determine if new responses along with harmonies will be acceptable. For using this memory more efficiently, there is a parameter that is introduced in order to show the choice harmony memory HMCR. The next factor has been the regulation of the sound volume that has some parameters of regulation and change of volume PAR. The parameters that follow have been used to create another new harmony [15]. During the iteration, new harmony vectors are generated by updating the memory consideration and pitch adjustment. Eq. (4) gives the value of each component to be updated in the memory consideration:

$$x_i^{new} = \begin{cases} x_i^j, j \in (1, 2, \dots, HMS), rand_i < HMCR \\ x_i \in X_i, else, \end{cases} \quad (4)$$

The pitch adjustment is carried out using Eq. (5):

$$x_i^{new} = \begin{cases} x_i^j \pm rand_2 * BW, rand_2 < PAR(\text{when } x_i \text{ is continuous}) \\ x_i(k+m), rand_2 < PAR(\text{when } x_i \text{ is discrete}) \\ x_i^{new}, \text{ else} \end{cases} \quad (5)$$

### 3.5 Proposed Hybrid Harmony – Tabu Replication Algorithm

This Harmony search algorithm includes both intensification and diversification [16]. The former will be controlled by a pitch adjustment and helps the local solutions to get retrained. There is a random selection that explores the global search space widely and efficiently. The pitch adjustment further makes this new solution good enough to the currently existing solutions. The latter is controlled by the consideration of memory resulting in a process of search within good solutions. This hybrid HS-TS algorithm has two phases: (a) The TABU Search algorithm and (b) the Harmony Search. The process employed in this algorithm is depicted in Fig. 4, [17,18].

1. Initialising of the population using random solutions
2. **Evaluation of the population's fitness**
3. When (the stopping criterion has not been met) // TABU Search Algorithm
  - 1: Choosing of an initial solution  $i$  in the  $S$ . Set the  $i^* = i$  and  $k=0$ .
  - 2: Set the  $k=k+1$  and further generate a new subset  $V$  of the solution in  $N(i,k)$  so that at least one of the TABU conditions is violated or one of the conditions of aspiration holds. 8
  - 3: Choosing the best  $j$  in tje  $V$  and set  $i = j$ .
  - 4: In case the  $f(i) < f(i^*)$  then set the  $i^* = i$ .
  - 5: Updating of the TABU and the conditions of aspiration.
  - 6: In case the stopping condition is arrived at, stop. If not, move to Step 2.
- // The Harmony Search Algorithm
7. Initializing of a Harmony Memory (HM).
8. Improvising a new harmony from the HM.
9. In case its new harmony is found to be better compared to the worst harmony of HM put the new one and exclude the worst one from the HM.
10. In case the stopping criterion is not satisfied, move to 8.

**Figure 4:** Pseudo code of the harmony search algorithm

## 4 Results and Discussion

The performance of the proposed Harmony-Tabu Replication algorithm is evaluated and compared with Simple Replication Algorithm, Tabu replication and Harmony Replication. CloudSim platform is used for simulating the environment. The number of datacenter and data to be replicated are varied for evaluating various scenarios. The focus of evaluation is percentage in bandwidth savings. Three different scenarios are considered, namely, performance when number of data centers are varied, number of data to be replicated are varied and for different capacity of the data centers. Tab. 1 shows the parameter of Tabu search and Harmony search.



**Table 1:** Simulation parameters

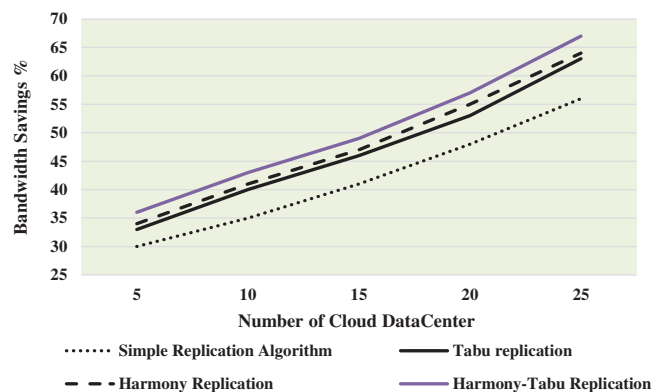
Parameters	Values	Parameters	Values
Maximum number of Iterations (Tabu Search)	20	Maximum number of Iterations (Harmony Search)	100
Length of Tabu Lists	96	HMCR	0.8
Length of candidate lists	48	PAR	0.75
Tabu Length	10		

**4.1 Number of Cloud Data Centres**

The impact of data centres has on replication is evaluated on the basis of bandwidth savings. [Tab. 2](#) and [Fig. 5](#) shows the bandwidth savings % with several cloud datacentre, in this work, the number of data centres investigated are 5, 10, 15, 20 and 25. It is observed from [Tab. 2](#) and [Fig. 5](#) that as the number of data centres increase, the bandwidth saving is higher. This is due to the fact the number of replicas which can be distributed around strategically is better, thus, leading to bandwidth savings.

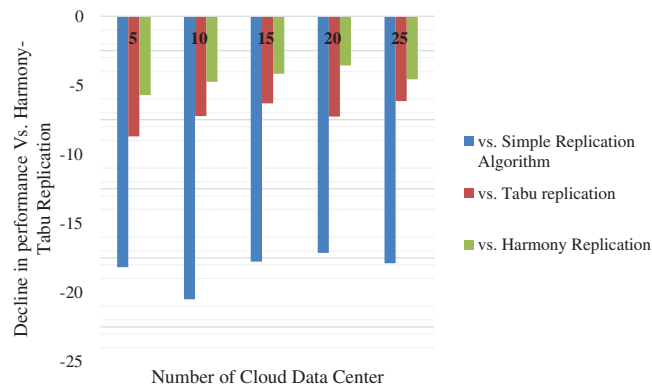
**Table 2:** Bandwidth savings (%) with number of cloud datacentre

Number of cloud data center	Simple replication algorithm	Tabu replication	Harmony replication	Harmony-Tabu replication
5	30	33	34	36
10	35	40	41	43
15	41	46	47	49
20	48	53	55	57
25	56	63	64	67



**Figure 5:** Bandwidth savings (%) with number of cloud datacentre

[Fig. 6](#) shows the decline in performance of simple replication, Tabu replication and Harmony replication compared to the proposed Harmony-Tabu replication.



**Figure 6:** Decline in performance for number of data centres compared to proposed harmony-tabu replication

It is seen that the bandwidth saving for proposed harmony-Tabu replication performs better by 18.18%, by 20.51%, by 17.78%, by 17.14% and by 17.89% at number of cloud datacenter 5, 10, 15, 20 and 25 respectively than simple replication algorithm. Similarly, the bandwidth saving for proposed harmony-Tabu replication performs better by 8.69%, by 7.23%, by 6.32%, by 7.27%, and by 6.15% at number of cloud datacenter 5, 10, 15, 20, and 25 respectively than Tabu replication. The bandwidth saving for proposed harmony-Tabu replication performs better by 5.71%, by 4.76%, by 4.17%, by 3.57%, and by 4.58% at number of cloud datacenter 5, 10, 15, 20, and 25 respectively than Harmony replication.

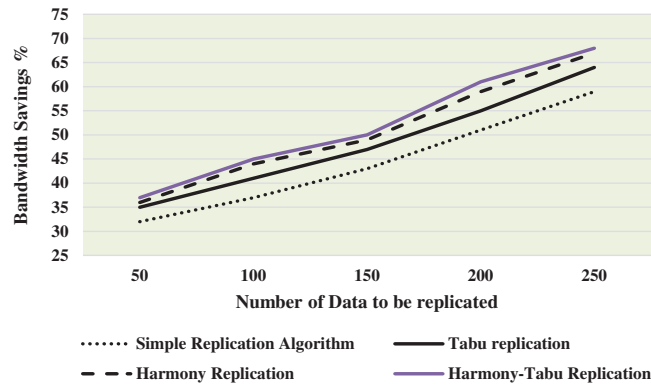
#### 4.2 Number of Data to be Replicated

The impact of number of data to be replicated is evaluated on the basis of bandwidth savings. [Tab. 3](#) and [Fig. 7](#) shows the bandwidth savings as the number of replicas varies, in this work, the number of data to be replicated investigated are 50, 100, 150, 200 and 250. It is observed from [Tab. 3](#) and [Fig. 7](#) that as the number of data replicas increase, the bandwidth saving is higher. As the access of the data is faster, thus, leading to bandwidth savings.

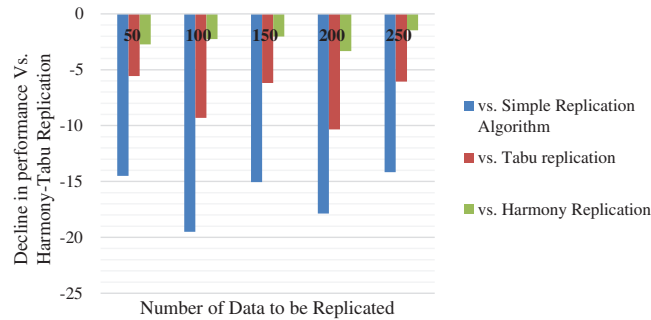
**Table 3:** Bandwidth savings (%) with number of data to be replicated

Number of Data to be replicated	Simple replication algorithm	Tabu replication	Harmony replication	Harmony-Tabu replication
50	32	35	36	37
100	37	41	44	45
150	43	47	49	50
200	51	55	59	61
250	59	64	67	68

[Fig. 8](#) shows the decline in performance of simple replication, Tabu and Harmony replication compared to the proposed Harmony-Tabu replication for varying number of data to be replicated. From [Tab. 3](#) and [Fig. 8](#), it is seen that the bandwidth saving for proposed harmony-Tabu replication performs better by 14.49%, by 19.51%, by 15.1%, by 17.86%, and by 14.17% at number of data to be replicated 5, 10, 15, 20 and 25 respectively than simple replication algorithm. The bandwidth saving for proposed harmony-Tabu replication performs better by 2.74%, by 2.25%, by 2.02%, by 3.33%, and by 1.48% at number of data to be replicated 5, 10, 15, 20 and 25 respectively than Harmony replication.



**Figure 7:** Bandwidth savings (%) with number of data to be replicated



**Figure 8:** Decline in performance for number of data to be replicated compared to proposed harmony-tabu replication

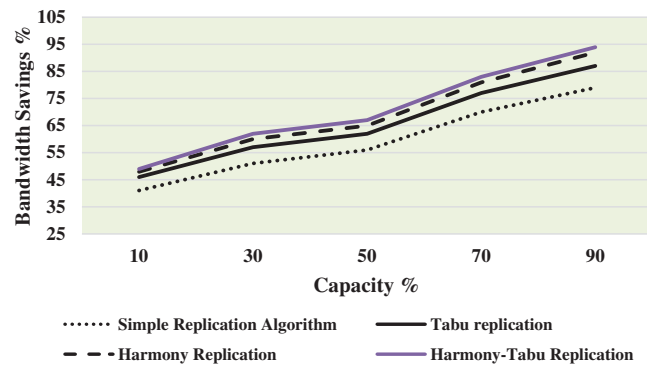
### 4.3 Capacity

The impact of capacity is evaluated on the basis of bandwidth savings. Tab. 4 and Fig. 9 shows the bandwidth savings as the capacity varies between 10 to 90. It is observed from Tab. 4 and Fig. 9 that as the capacity increases, the bandwidth saving is higher.

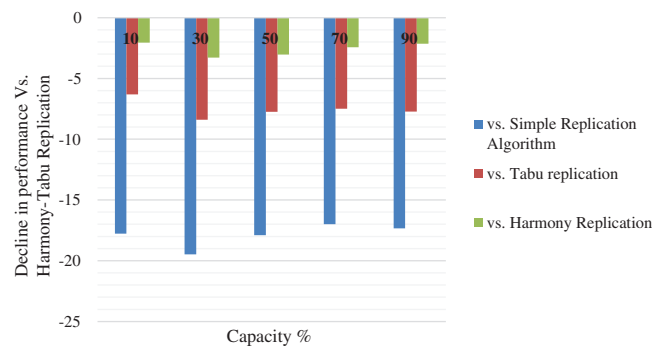
**Table 4:** Bandwidth savings (%) with capacity %

Capacity %	Simple replication algorithm	Tabu replication	Harmony replication	Harmony-tabu replication
10	41	46	48	49
30	51	57	60	62
50	56	62	65	67
70	70	77	81	83
90	79	87	92	94

Fig. 10 shows the decline in performance of simple replication, Tabu replication and Harmony replication compared to the proposed Harmony-Tabu replication for varying capacity.



**Figure 9:** Bandwidth savings with capacity %



**Figure 10:** Decline in performance for varying capacity compared to proposed harmony-tabu replication

From [Tab. 4](#) and [Fig. 10](#), it is seen that the bandwidth saving % with capacity% for proposed harmony-Tabu replication performs better by 17.78%, by 19.47%, by 17.89%, by 16.99%, and by 17.34% at capacity % 10, 30, 50, 70 and 90 respectively than simple replication algorithm. Similarly, the bandwidth saving % with capacity% for proposed harmony-Tabu replication performs better by 6.32%, by 8.4%, by 7.75%, by 7.5%, and by 7.73% at capacity% 10, 30, 50, 70, and 90 respectively than Tabu replication. The bandwidth saving for proposed harmony-Tabu replication performs better by 2.06%, by 3.28%, by 3.03%, by 2.44%, and by 2.15% at number of data to be replicated 10, 30, 50, 70, and 90 respectively than Harmony replication.

It is observed from the experimental results that the proposed method saves significantly in bandwidth when compared to other methods. But, the main limitation of the proposed algorithm is that the cost and energy are not considered during the optimization process.

## 5 Conclusion

Data replication in the cloud is used to reduce waiting time by users and bring down the bandwidth consumption by offering multiple replicas to the user for various nodes. The replication of data further offers a practical solution to the issue by maintaining the copies replicated in the sites closer to the consumers of data that reduce both the time and the bandwidth consumption. The metaheuristics are widely used for optimizing the data replication process. The work proposed a hybrid Tabu Search using Harmony Search in order to perform replication of data effectively. The Tabu Search makes use of short-term as well as long-term memory while at the same time making moves between all the neighbouring solutions. This is quite essential for the local search to remain balanced in terms of their quality and time

taken for computing. A hybrid algorithm Harmony-Tabu is proposed to improve the performance of optimization. The results have proved that a percentage of bandwidth saving for a varying number of cloud datacentres for the proposed replication of Harmony- TABU has performed better compared to a simple algorithm of replication, Tabu replication and Harmony replication. Future work must investigate the scalability of the proposed algorithm. Parameters like energy and security can be investigated.

**Acknowledgement:** Thanks to the authors and reviewers.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] Y. Ebadi and N. Jafari Navimipour, "An energy-aware method for data replication in the cloud environments using a tabu search and particle swarm optimization algorithm," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 1, pp. 47–57, 2018.
- [2] W. Li, Y. Yang, J. Chen and D. Yuan, "A cost-effective mechanism for cloud data reliability management based on proactive replica checking," *IEEE/ACM Int. Symposium. Cluster, Cloud and Grid Computing (CCGrid)*, vol. 1, no. 1, pp. 564–571, 2012.
- [3] P. J. Kumar and P. Ilango, "BMAQR: Balanced multi attribute QoS aware replication in HDFS," *International Journal of Internet Technology and Secured Transactions*, vol. 8, no. 2, pp. 195–208, 2018.
- [4] S. K. Sandhu and A. Kumar, "Hybrid meta-heuristics based scheduling technique for cloud computing environment," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 1457–1465, 2017.
- [5] M. Amutha and R. Sugumar, "A survey on dynamic data replication system in cloud computing," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, no. 4, pp. 1454–1467, 2015.
- [6] N. Mansouri and M. M. Javidi, "A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers," *Journal of Supercomputing*, vol. 74, no. 10, pp. 5349–5372, 2018.
- [7] M. Shorfuzzaman, "Access-efficient qos-aware data replication to maximize user satisfaction in cloud computing environments," *2014 15th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, vol. 1, no. 1, pp. 13–20, 2014.
- [8] J. W. Lin, C. H. Chen and N. M. Chang, "Qos-aware data replication for data-intensive applications in cloud computing systems," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 101–115, 2013.
- [9] T. Loukopoulos and I. Ahmad, "Static and adaptive distributed data replication using genetic algorithms," *Journal of Parallel and Distributed Computing*, vol. 64, no. 11, pp. 1270–1285, 2004.
- [10] Y. Ebadi and N. Jafari Navimipour, "An energy-aware method for data replication in the cloud environments using a Tabu search and particle swarm optimization algorithm," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 1, pp. 47–57, 2019.
- [11] A. Awad, R. Salem, H. Abdelkader and M. A. Salam, "A novel intelligent approach for dynamic data replication in cloud environment," *IEEE Access*, vol. 9, pp. 40240–40254, 2021.
- [12] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement," *Frontiers of Computer Science*, vol. 10, no. 5, pp. 925–935, 2016.
- [13] A. Vishnu Priya and S. Ashish kumar, "Hybrid optimal energy management for clustering in wireless sensor network," *Computers and Electrical Engineering*, vol. 86, pp. 379–387, 2020.
- [14] M. Armaghan, A. T. Haghighat and M. Armaghan, "Qos multicast routing algorithms based on tabu search with elite candidate list," *Application of Information and Communication Technologies, International Conference*, vol. 1, no. 1, pp. 1–5, 2009.

- [15] O. Norouzpour and N. Jafarzadeh, "Using harmony search algorithm for load balancing in cloud computing," *Indian Journal of Science and Technology*, vol. 8, no. 23, pp. 1–4, 2015.
- [16] S. Thangavel, T. V. Rajeevan, S. Rajendrakumar, P. Subramaniam, U. Kumar *et al.*, "A collaborative mobile based interactive framework for improving tribal empowerment," *Advances in Intelligent Systems and Computing*, vol. 1108, pp. 827–842, 2020.
- [17] X. S. Yang, "Harmony search as a metaheuristic algorithm," *Music-Inspired Harmony Search Algorithm*, Springer, vol. 1, no. 1, pp. 1–14, 2009.
- [18] K. Pradeep and T. P. Jacob, "A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment," *Wireless Personal Communications*, vol. 101, no. 4, pp. 2287–2311, 2018.