Received January 16, 2021, accepted February 14, 2021, date of publication March 9, 2021, date of current version March 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064917

A Novel Intelligent Approach for Dynamic Data Replication in Cloud Environment

AHMED AWAD[1], RASHED SALEM[2], HATEM ABDELKADER[2], AND MUSTAFA ABDUL SALAM[3]

[1]Information Systems Department, Cairo Higher Institute for Languages and Simultaneous Interpretation, and Administrative Science, Cairo 11571, Egypt
[2]Information Systems Department, Faculty of Computers and Information, Menoufia University, Shebeen El-Kom 32511, Egypt
[3]Artificial Intelligence Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

Corresponding author: Mustafa Abdul Salam (mustafa.abdo@fci.bu.edu.eg)

ABSTRACT In recent years, cloud computing research, specifically data replication techniques and their applications, has been growing. If the replicas number is raised and put in multiple positions, it will be expensive to maintain the data usability, performance and stability of the application systems. In this paper, two bio- inspired algorithms were proposed to improve both selection and placement of data replicas in the cloud environment. The suggested algorithms for dynamic data replication are multi-objective particle swarm optimization (MO-PSO) and ant colony optimization (MO-ACO). The first suggested algorithm, *i.e.*, MO-PSO, is employed to obtain the best selected data replica depend on the most frequent one. However, the second suggested algorithm, *i.e.*, MO-ACO, is employed to obtain the best data replica placement depend on the shortest distance, and the replicas availability. A simulation of the suggested strategy was carried out using CloudSim. Each data center (DC) includes hosts with set of virtual machines (VMs). The data replication order is determined at random from a thousand cloudlets. All replication files are randomly distributed in the proposed architecture. The performance of suggested techniques was evaluated against several approaches including: Adaptive Replica Dynamic Strategy (ARDS), Enhance Fast Spread (EFS), Genetic Algorithm (GA), Replica Selection and Placement (RSP), Popular File Replication First (PFRF), and Dynamic Cost-aware Re-replication and Re-balancing Strategy (DCR2S). The simulation results prove that MOPSO gives improved data replication compared against other algorithms. Additionally, MOACO realizes higher data availability, lower cost, and less bandwidth consumption compared with other algorithms.

INDEX TERMS Dynamic replication, cloud environments, CloudSim, multi objective optimization, particle swarm optimization and ant colony optimization.

I. INTRODUCTION

Cloud environments provide several services, such as pay-per-use virtual computing and network resources. The cloud's characteristics include storage, bandwidth, and access to valuable data resources [1], [2]. A cloud environment incorporates infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) [3], [4]. Furthermore, the cloud is cheaper compared with traditional systems and offers benefits such as subscription-based billing, scalability, dynamicity, and elasticity [5], [6]. Load balancing in the cloud environment is very important to achieve best performance [7]–[9]. It allocates tasks to realize optimal access to available resources [10]–[15].

The associate editor coordinating the review of this manuscript and approving it for publication was Huiling Chen.

Data-intensive applications, including sharing and dissemination from distant points to nearby points through nodes or geographical sites, commonly employ replication techniques [16], [17]. In a cloud system, replication methods can be used as a cluster. The cluster ensures the accuracy and integrity of different replications within nodes by allowing for data availability and scalability. This procedure entails reading and writing on data replica using protocols that are specific to the device [18]–[21]. In [22]–[25], there are reviews that present static and dynamic replications. Indeed, both static and dynamic replications are facing three significant questions that need to be answered: 1) what data should be replicated? 2) when the data should be replicated? and 3) where the new replicas should be placed? These three main open questions require to be tackled for data replication in the cloud environments [26]–[31].

A. MOTIVATION

In data processing, data replication is one of the most vital subjects. However, the implementation of applications in cloud still gives low reliability and efficiency. If the replicas number is increased and put in various locations, it would be expensive to maintain the data performance, availability, reliability of a system.

Data replication is expressed as a multi-objective optimization problem and swarm-intelligence (SI) algorithms were applied for dynamic data replication. SI algorithms are applied to discover the optimal solutions. These algorithms also give better results than the static replication methods.

B. CONTRIBUTIONS

In this paper, intelligent dynamic data replication algorithms are proposed based on bio-inspired algorithms with multi-objective (MO-PSO, and MO-ACO). The introduced strategies are used for both data replicas selection and placement in various datacenters. The introduced algorithms are tested using CloudSim. The performance of suggested techniques were evaluated against several replication strategies including, Adaptive Replica Dynamic Strategy (ARDS), Enhance Fast Spread (EFS), Genetic Algorithm (GA), Replica Selection and Placement (RSP), Popular File Replication First (PFRF), and Dynamic Cost-aware Re-replication and Re-balancing Strategy (DCR2S). The experimental results illustrate that MOPSO reaches improved data replication compared with other algorithms. Furthermore, MOACO realizes lower cost, less bandwidth consumption, and higher data availability compared with other techniques.

C. PAPER ORGANIZATION

The rest of this paper is organized as follows. Section II is devoted for related work. Section III presents the suggested approach for both replica selection and placement. Moreover, section IV presents the two proposed algorithms. Section V illustrates the simulation configuration. While section VI discusses and analyzes the results, section VII addresses the performance evaluation measures. Finally, section VIII concludes the suggested work and highlights the future work.

II. RELATED WORK

Many relevant studies in the literature addressed data replication techniques in the cloud environment, as follows:

N. Mansouri *et al.* presented a dynamic replication approach access data applying the 80/20 idea. The introduced approach was evaluated with ADRS and RSP algorithms, and achieved best accuracy [32].

D. Sun *et al.* introduced a dynamic data replication strategy (MDDRS). The numbers of newly placed replications were effectively determined and allocated in parallel between DCs using the mathematical model in CloudSim. Moreover, the study carried out a review of the proper distribution approach in the cloud environment [33].

N. Kaur and *et al.* suggested an enhanced data replication approach. The suggested approach is an enhanced version of the research in [33] and they decided the minimum data replicas and file availability adjacent the users in an ordinary tier without missing data [34].

N. Mansouri illustrated a model called ADRS for replica placement and replacement via datacenters. The ADRS was evaluated against five algorithms, *i.e.*, ARS, DRSP, CIR, CDRM and build-time strategies; the outcomes proved its ideal proficiency over those of the other algorithms [35].

K. Kumar *et al.* suggested a data replication technique to optimize consumption of resources in cloud environments. Furthermore, the proposed solution reduces the total time for a search request or another procedure, and it shares structures in the device context for analyzing work-loads and storing data utilizing mathematical models [36].

X. Bai *et al.* introduced a response time-based replica management (RTRM) model that selects replicas and allocates them in nodes automatically. The model is constrained by the limited response time when applied to heterogeneous HDFS platforms. If the time exceeds the specified threshold, the RTRM algorithm will be automatically increased upon the user's request [37].

D. Yuan *et al.* proposed a technique for data placement. It was depending on a k-means clustering algorithm. The presented strategy dynamically moves to the best datacenters via the network. When there are no more suitable places for the small data collected while completing a mission, the technique looks for other cloud sites [38].

C. Hamdeni *et al.* suggested a novel method for estimating the popularity in databases via nodes. To access data replicas using the different data popularity methods in new replication management contexts, one of the proposed methods is the access time, which has a dynamic nature in the underlying system. The data popularity approach analyzes the historical data access or most currently data that is easily accessible and can be measured by the replication management system. The simulation results appeared that the suggested strategy can adjust the most common file access time and realize file availability [39].

N. Maheshwari *et al.* proposed a dynamic data placement approach using GridSim. It preserves the power consumption among clusters. The suggested model achieved effective power availability in the development of distributed data [17].

Q. Xu *et al.* introduced a data placement strategy based on a genetic algorithm (DPSBGA) to improve the data placement via datacenters. It reduces the scheduling processes among the datacenters [40].

Z. Li *et al.* introduced a replication method to improve the high-level architecture performance. Different replications are developed in the underlying approach. The proposed structure employed various synchronization approaches for rapid replications [41].

Z. Tang *et al.* suggested a load balancing data placement method in a Spark environment for optimizing the load balancing among the nodes. The proposed algorithm handled the data in all clusters, reduced the time of execution and balanced the tasks among nodes [42].

T. Yuan *et al.* suggested two methods for both partially and fully replicated systems. For the purpose of executing causal consistency tasks in stable large-scale distributed networks, the two approaches realized the causal consistency of partial and absolute replications [43].

H. Casanva *et al.* analyzed the replication process effects on an application that had large-scale concurrent executions and synchronized complete replication checkpointing. The authors further propose a new perspective on exponential failure distributions, which includes the average amount and average out time during regression cycles over a broad range of systems [44].

P. Matri *et al.* suggested a decentralized, write-enabled dynamic geo-replication (DWEDGR) approach. DWEDGR lets users to connect to the closest replication between nodes without having to make a prior request for data that can be changed [45].

M. Chang Lee *et al.* suggested the popular file replicate first (PFRF) method. Replications to suitable locations was adopted so that users could access suitable distributions using clusters. Both Zipf distributions and geometric distributions are employed to assess the suggested method. The replication placements can be dynamically changed at any time according to each file's popularity, and further analysis can be conducted to balance tasks [46].

N. Mansouri *et al.* introduced a replication method using fuzzy based self-defense algorithm. Six optimization objectives (*i.e.*, availability, service time, load, energy consumption, latency, and centrality) are used. This strategy has achieved good results in all mentioned objectives [47].

L.Chunlin *et al.* designed a dynamic multi-objective optimized replica placement strategy in edge cloud. The minimum data replicas, data transmission, file availability and unavailability are determined nearby the users. Results indicated that the suggested strategy achieved good results in all aspects [48].

Y. Ebadi *et al.* introduced an energy aware technique for data replication employing a tabu search and PSO. The numbers of newly placed replications were effectively determined [49].

A comparison has been presented for different replication techniques and strategies, like the replicas selection and placement among different nodes. Previous studies have several problems that should be investigated for replications in the cloud environment. Most of these strategies have disregarded the usage of AI with MOO. Our MO-PSO and MO-ACO algorithms have achieved optimal access to the selection and placement of replicas via nodes. Furthermore, replicas are accessed using the least-cost path as well as the maximum number of tasks by users.

A. DISCUSSION AND RELATED WORK

In TABLE 1, a comparison between previous studies in cloud computing has been presented in terms of its advantages and limitations. Most of previous studies have disregard multiobjective swarm intelligence algorithms. The limitations of this work are also summarized. The consistency and the energy consumed were not addressed in this work.

III. THE PROPOSED DATA SELECTION AND PLACEMENT ARCHITECTURE

This section presents the proposed structural model for both selection and placement of replicas in cloud environments. To maximize the data replicas selection and placement, a heterogeneous method using swarm intelligence algorithms were employed. FIGURE 1 depicts the proposed scheme, which contains data centers placed in various stages. The high datacenters, which are more highly centralized and have better data access, storage capacity, output, and a greater number of hosts and VMs compared with other datacenters, are the first. The mid DC is the second stage, which has less entire components than the first stage. Low DC components make up the third level, which has fewer entire components than the second. Generally, the datacenters are linked to one another on a hierarchical basis, either on the same level or at higher levels. The proposed infrastructure consists of data centers, brokers, replica catalogs, replica management systems, tasks, hosts, VMs, and hierarchical network users that link data centers to each other. Applying MO-ACO, and MO-PSO techniques enhances the availability, reduces the cost, enables tasks to be completed faster than the algorithms that are utilized when selecting and placing replicas via CloudSim.

The datacenters are represented as $DCs = \{dc_1, dc_2, \ldots, d_{cn}\}$, where n is the count of DCs. The physical machine (PM) can be shown as $PM = \{pm_1, pm_2, \ldots, p_{mx}\}$, where x represents a set PMs in DCs. The VMs can be shown as $VM = \{vm_1, vm_2, \ldots, vm_s\}$, where s is the count of VMs that are allocated in the PMs. The data replication can be expressed as $F = \{f_1, f_2, \ldots, f_y\}$, where y is the count of replicas. The main storage unit can be expressed as $B = \{b_1, b_2, \ldots, b_y\}$, where y is a set of various data replicas that are stored in DCs. The replica files are stored in high data centers and uniformly distributed in the proposed technique. The replica catalog, which records each replica location in the various DCs, can be used to save different values of the $probability = pro(bap)$ in each DC.

A. DATA FILE AVAILABILITY

Availability can be described as ''readiness to provide correct services''. On request, all users should be able to access complete replications of the data. The data availability is also a critical problem for consumers of the cloud. A server with inaccessible data is triggered by lost data replication or network malfunction in the cloud [33], [34].

Replicas can be stored in several DCs, and multiple replicas can be allocated in the same DC to guarantee the presence of multiple DC blocks. High DCs come at higher costs, improved storage access and stability and, as a result, blocks inside DCs achieves high availability. However, low DCs

TABLE 1. Comparison of the reviewed data replication techniques in the cloud computing.

Technique	Advantages	Limitations
SWORD	• High availability • High load balancing	• High latency • High replication cost • Low reliability • High response time
D2RS	• High availability • Low bandwidth • Low waiting time	• Low-speed data access • High user waiting time • Low load balancing • High replication cost
MORM	• Low storage cost • High availability • High reliability	• High bandwidth • High replication cost
Fuzzy-FP	• High availability • High performance	• High response time • High replication cost
EIMORM	• High performance • High availability • Low bandwidth	• High replication cost • High response time
DPRS	• High availability • High data durability • Low bandwidth	• High energy consumption • High storage cost • High response time • High replication cost
PDR	• High availability • High data durability • Low bandwidth • Low response time	• High replication cost • Low consistency rate
EFS	• Medium load balancing • High availability • High data durability	• Low reliability • High energy consumption • High storage cost
DCR2S	• Low replication cost • High availability • High reliability	• High response time • High time complexity • Low consistency rate
ACO	• High utilization rate • Low response time	• Without considering the cost and consumed energy
GASA	• High availability • High reliability	• High response time • High replication cost
DRSACO	• High reliability • High availability • High scalability	• High time complexity • High energy consumption
Genetic	• Low bandwidth • High availability	• High update rate • High replication cost
MOGA	• High scalability • High performance • High availability	• High replication cost • High bandwidth • High energy consumption
Proposed model	• High availability • High performance • High reliability • Low bandwidth • Low response time • Low replication cost • Low storage cost • High load balancing • Speed access data • Least-cost path • Low distance to access node • Optimal location	• Consistency and energy consumption are not considered.

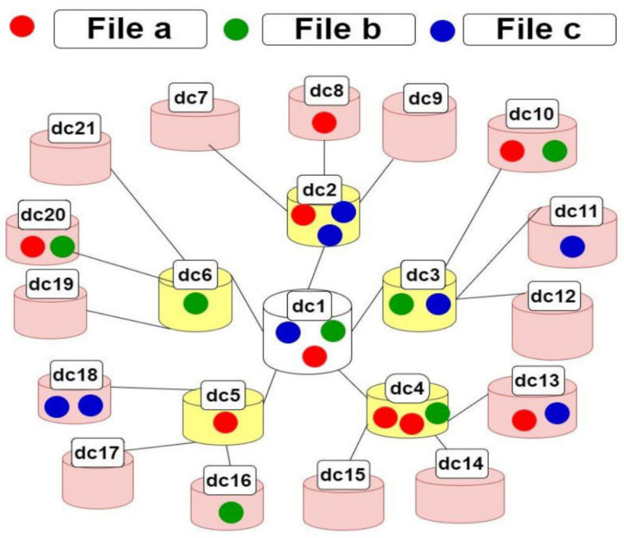


FIGURE 1. The proposed data selection and placement architecture.

come at lower costs, worse reliability, and bad availability. These DCs can be calculated using Eq. (1) to Eq. (4) as follows:

$$pro(bap_j)_{high_{DC}} > pro(bap_j)_{mid_{DC}} > pro(bap_j)_{low_{DC}} \quad (1)$$

$$pro(fla_k) = \begin{cases} (1 - \prod_{i=1}^{bnr_k}(1 - pro(bap_j)i))^{nb_k} & \text{case 1} \\ \prod_{i=1}^{nb_k}(1 - \prod_{i=1}^{bnr_k}(1 - pro(bap_j)i)) & \text{case 2} \end{cases} \quad (2)$$

$$\overline{pro(fla_k)} = \begin{cases} 1 - (1 - \prod_{i=1}^{bnr_k}(1 - pro(bap_j)i))^{nb_k} \\ 1 - \prod_{i=1}^{nb_k}(1 - \prod_{i=1}^{bnr_k}(1 - pro(bap_j)i)) \end{cases} \quad (3)$$

Let the block available probability $pro(bap_j)$:

$$high_{DC} = 0.9 > mid_{DC} = 0.6 > low_{DC} = 0.3 \quad (4)$$

where

$pro(bap_j)$	Block availability probability
b	Blocks
$pro(fla_k)$	File availability Probability
nb_k	Number of block
bnr_k	Number of replica of a data file df_k
$\overline{pro(fla_j)}$	Block unavailability probability
na_k	Number of access task have request
$\overline{pro(fla_k)}$	File unavailability probability
$high_{DC}$	High datacenter
mid_{DC}	Medium datacenter
low_{DC}	Low datacenter

B. DATA REPLICATION OPTIMIZATION IN CLOUD COMPUTING

Cloud computing using DCs consists of replicas and PMs, VMs with certain storage spaces, distances, times and every replica cost of the DCs. To achieve the contradicting targets, including time, costs and space, for optimizing data replication access; both selection and placement are considered two of the most significant issues that face users in the cloud. The target with the least expensive and shortest path for data replication should be selected to reach high availability.

1) REPLICATION COSTS IN DATACENTERS

There is a set of DC costs that distinguish low DCs, mid DCs, and high DCs. This set involves prices, availability probabilities, speeds, reliability and cloud performance. The DCs' costs are essential factors that contain the MOO factors that pick and position replicas over DCs. The replication total cost must be held as low as possible in order to minimize replication through DCs and guarantee that the budget is adequate for users.

$$cost_k(dc_s) = \sum_{x=1}^{y}(cost(dc_y) * bnr_k(dc_y)) \tag{5}$$

2) MINIMUM DISTANCE BETWEEN DATACENTERS

The distance among DCs is determined on the basis of adata replicas' access, which can be measured using Eq. (6) and Eq. (7). The best solution exploits the link between two DCs (i and j). By confirming that this equation ensures their non-passing in an infinite ring, this equation ensures that d_{ci} and d_{cj} pass.

$$Min \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \tag{6}$$

$$\text{S.T: } \sum_{i=1}^{n} n_i x_{i \geq k}, \quad x_i \in \{0, 1\}, (1 \leq i \leq n) \tag{7}$$

3) KNAPSACK PROBLEM

The knapsack is considered an NP-hard problem. Any object has its allocated value and weight. The goal is to reduce cloud costs by guaranteeing that the budget is appropriate for consumers by using Eq. (8) and Eq. (9):

$$\text{maximize} \quad px = \sum_{j=1}^{n} p_j x_j \tag{8}$$

$$\text{S.T: } wx = \sum_{j=1}^{n} w_{ij} x_j \leq v_i \tag{9}$$

where
$x_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n$
$p = (p_1, p_2, \ldots, p_n)$
$w = w_1, w_2, \ldots, w_n$
$i = 1, 2, \ldots, m$

Each object $j \in J$ has profit p_j and weight w_j in dimension i, where $(1 \leq i \leq m)$. Binary variable x_j indicates whether object j is included in the knapsack ($x_j = 1$) or not ($x_j = 0$).

TABLE 2 shows the parameters of knapsack algorithm, weights (w) and values (v). The algorithm determines what is accepted and rejected according to the files placed and takes into consideration the cost of each file on the datacenter. Thus, the main three data file replicas need to be replicated on mid and lower cost data centers, *i.e.*, the low datacenters in the path of maximum tasks from users. MO-PSO and MO-ACO are employed to estimate the data availability and optimal costs in DCs.

TABLE 2. Knapsack algorithm parameters.

i	w_i	v_i
1	500	5
2	300	1
3	300	1
4	100	10
5	100	10
6	100	10
7	300	15
8	100	15

IV. PROPOSED METHODS FOR DATA REPLICATION

The proposed scheme combines two algorithms: PSO and ACO. PSO will pick the data replicas and ACO will position the data replicas. The proposed functional structure therefore has three fundamental characteristics: availability of files, time of access and costs. Zipf and the geometric distribution are utilized to spread the data replicas in the cloud.

A. THE PROPOSED STRATEGY USING PSO TO DETERMINE WHICH FILE TO REPLICATE AND WHEN TO REPLICATE

In this subsection, we introduce MO-PSO algorithm for data replication using CloudSim. For each particle with the best data replica, a fitness function tests the optimally chosen data replication. The velocity, position and inertia weight updates are shown in Eq. (10), Eq. (11) and Eq. (12) [50]–[55] as follows:

$$V_{i,j}^{k+1} = W.V_{i,j}^{k} + C_1R_1(pbest_{i,j}^{k} - X_{i,j}^{k}) + C_2R_2(gbest_{i,j}^{k} - X_{i,j}^{k}) \tag{10}$$

where

$V_{i,j}^{k+1}$	particle's new velocity
$V_{i,j}^{k}$	particle's current velocity
C_1, C_2	positive constants acceleration parameters
$pbest_{i,j}^{k}$	personal best position particle
$X_{i,j}^{k}$	position of i^{th} particle in j^{th} swarm
R_1, R_2	random numbers in the range [0,1]
$gbest_{i,j}^{k}$	global best position particle

$$X_{i,j}^{k+1} = X_{i,j}^{k} + V_{i,j}^{k+1} \tag{11}$$

where

$X_{i,j}^{k+1}$	new position of particle
k	iteration population
$i \in 1, 2, 3, \ldots, m$	m is members number in an iteration
$j \in 1, 2, 3, \ldots, d$	d is the size of the swarm

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter \quad (12)$$

where

w	inertia weight
w_{max}	max value of inertia weight
w_{min}	min value of inertia weight
$iter_{max}$	maximum number of iterations
$iter$	current iteration number

The proposed MO-PSO algorithm is introduced in (Algorithm 1). The fitness function is shown in Eq. (13) to Eq. (15).

Algorithm 1: The Proposed MOPSO for Selecting Data Replicas on Cloud Environment

```
Input: Size $\alpha$ of population, Number of iterations,
       Datacenters, Data availability, Improved
       Time-Based Decay Function
Output: Selected $P_{position} \leftarrow$ (Optimal$_{BestReplica}$,
        Optimal$_{TotalExecutionTime}$ and Optimal$_{Costs}$)
begin
    No. of iterations and No. of particles
    Initialize set values of particle swarm (iteration)
    Initialize un / availability probabilities
    Initialize replicas according to costs and time
    repeat
        for $j = 1$ to $\alpha$ do
            foreach data replication in DC
            do
                Calculate fitness function
                Update velocity
                Update position
                $P_{velocity} \leftarrow$ Random velocity()
                $P_{position} \leftarrow$ Random position()
                $P_{best} \leftarrow P_{position}$
                if $\alpha \leq 0$ then
                    Exploitation
                else
                    Exploration
                    Select best data replication
                Calculate the ITBDF
                Calculate the replica factor
                Calculate the costs
    until reaching the maximum number of iterations,
      or finding access solution
    Return the optimal best replica solution
```

B. FRAGMENT TYPE EVALUATION AND ACCESS TIME ESTIMATION

In DCs with varying accesses and data intervals, the ITBDF is utilized to assess priorities and data replication weights. The ITBDF weights access to latest data replicas and separates previously accessible data for this concept. Based on the proposed MO-PSO algorithm, which is applied in our framework, the ITBDF assigns priority and weight for replicas. The ITBDF is expressed in Eq. (13) and Eq. (15):

$$ITBDF(t_a, t_b) = e^{-(t_a - t_b)k} \quad \forall \quad k \in \{1, 2, 3, \ldots, n\} \quad (13)$$

$$ITBDF(t_a, t_b) = e^{-(\triangle t)k} \quad (14)$$

where $\triangle t = (t_a - t_b)$

where

t_a	current time
t_b	start time
k	step value
e	exponential function decay.

$$RF_k = \frac{\sum_{t_i=t_b}^{t_a} (na_k(t_i, t_i + 1) * ITBDF(t_i, t_a))}{bnr_k * \sum_{i=1}^{nb_k} sb_i} \quad (15)$$

where

na_k	the No. of accesses
bnr_k	the No. of replicas
nb_k	the No. of blocks
sb_i	the size of a block

C. THE PROPOSED MO-ACO FOR PLACEMENT

The usage of the proposed MO-ACO algorithm on a large scale to implement the placement in CloudSim is discussed in this paragraph. Replica management determines new replica placements based on replication costs as well as datacenter space at this level. After that, udata replicas are accessible to users. The proposed MO-ACO algorithm is employed to optimally place replicas in DCs to satisfy user requests.

The objective function is employed to calculate the cumulative pheromones at various points in the paths, the shift from DC_i to DC_j is estimated as shown in Eq. (16) to Eq. (21) [56], [57]:

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in k} [\tau_{is}]^\alpha [\eta_{ij}]^\beta} \\ 0 \quad Otherwise \quad if\ j \in allowed\ k \end{cases} \quad (16)$$

The chosen DC is estimated according to Eq. (17):

$$j = \begin{cases} argmax_{s \in allowed\ k} \{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta\}, & if q \leq q_0, if\ q > q_0 \\ j \end{cases} \quad (17)$$

The ant's detection array is calculated in accordance with Eq. (18):

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (18)$$

After each repetition, the pheromone values on the routes are changed. The pheromone value is a positive constant as ants near the end of their travel line. Eq.(19) can be used to measure the modified local pheromone value:

$$\tau_{ij} = (1 - p)\tau_{ij} + p_{\tau_0}, \forall (i, j) \in t_k, \text{where}(0 < p \leq 1) \quad (19)$$

TABLE 3. Notation of ant colony optimization (ACO).

Symbol	Description
α	Pheromone trial control parameter τ_{ij}
β	Pheromone trial control parameter η_{ij}
s	Probability for slave ant $\forall s \in \{1, 2, 3, \ldots, k\}$
k	Ants
q	Random variable uniformly distributed in [0,1]
i	Currently replica i
j	Choose go to replica j
p	Evaporation rate
m	Number of ants
τ_{ij}	Pheromone density (amount of pheromone between $i\&j$)
η_{ij}	Heuristic information (importance between node $i\&j$)
p_{ij}	Probability ant transits. Currently replica i to replica j
$p_{\tau 0}$	Initial pheromone
$\Delta_{\tau_{ij}}$	Amount of pheromone that an ant adds to the visited path
c^k	The length of tour t^k which was built by the ant k
t^k	Total lengths of path
t_k	Tour by ant k iteration
d_{ij}	Distance between two nodes

After evaporation, each ant contributes pheromones to the routes using the fixed process, and Eq. (20)adjusts the modified global pheromone value:

$$\tau_{ij} = (1-p) + p. \sum_{k=1}^{m} \Delta_{\tau_{ij}} \tag{20}$$

$\Delta_{\tau_{ij}}$: is the amount of pheromone applied to an ant's path, as shown in Eq. (21):

$$\Delta_{\tau_{ij}} = \begin{cases} \frac{1}{c^k} & if \forall (i, j) \in t^k \\ 0 & otherwise \end{cases} \tag{21}$$

Depending on geometric and Zipf distributions, the proposed MO-ACO algorithm positions and accesses optimal replicas. The fitness function is calculated for each MO-ACO of the replica positions in the cloud via DCs.

D. REPLICA SELECTION AND PLACEMENT WITH ZIPF AND GEOMETRIC DISTRIBUTIONS

To pick and position replicas in the cloud via DCs, Zipf and geometric distributions are employed. Users' activities are tracked by a distribution, which determines the most common data replicas and positions them in DCs closer to the users. A Zipf distribution is employed to model the scale at random, and the replica places are chosen based on their importance and ease of access:

$$p(f_i) = \frac{1}{i^\alpha} \tag{22}$$

where $i = 1, 2, \ldots, n$; and α is a data distribution factor, where $0 \leq \alpha < 1$.

With greater access and higher weights, a geometric distribution offers the most common data with a greater degree of distinction. The data can be spread using DCs, and the search space can be utilized to discover further random solutions as shown:

$$p(i) = (1-p)^{i-1}.p \tag{23}$$

where $i = 1, 2, \ldots, n$ and $0 < p < 1$. A higher p indicates that a smaller percentage of the data is accessed more frequently.

When users execute the replica control job, we decide that the algorithm accesses and places the most appropriate data node. The most powerful algorithm for finding the shortest and least cost path is MO-ACO.

The proposed MO-ACO algorithm is presented in (Algorithm 2). The fitness function is presented in "Eq. (16) to Eq. (23)".

Algorithm 2: The Proposed MOACO for Data Replica Placement in a Cloud Environment

```
Input: Population size, No. of iterations,Minimum
       distance between DCs
Output: Optimal Data Replica Placement
begin
    Define values of parameters, No. of iterations and
      No. of ants
    Initialize un / availability probabilities
    Initialize distance between DCs
    Initialize data replication costs and size
    Initialize optimal best data replica placement in DC
      solution
    repeat
        for I = 1 to (No. of ants) do
            Step = Step + 1
            Set all ant distribution in DC
            foreach DC in current system do
                Calculate desirability of the movement
                Calculate probability of the movement
                if $q \leq q_0$ then
                    Exploitation
                else
                    Exploration
            foreach dimension do
                Calculate fitness function
                Update local pheromone
                Update global pheromone
                Set local pheromone update
                Set global pheromone update
                Set replica placement in DC
                Until all replicas are selected
                Until all replicas are placed
                if the storage space of the DC is small
                  then
                    Apply the global update rule
                else
                    Delete small replica popularities
    until maximum number of iterations has been
      reached or access solution discovered
    Return the optimal data replica placement in the DC
```

V. SIMULATION CONFIGURATION

A. SWARM INTELLIGENCE EXPERIMENTS

The experimental findings are discussed in this section, and the replication method in the proposed cloud is designed. It also uses the proposed MO-PSO and MO-ACO and various distributions to position the replicas. CloudSim is utilized to run these algorithms. This method's execution time, speed of entry, costs, placement efficiency, and high replication availability are compared against other algorithms.

B. CONFIGURATION DETAILS

As illustrated in FIGURE 1, The cloud is created to represent various types of DCs with various structures as revealed in TABLE 4. Each DC is made up of a host and a series of VMs that include blocks of usable data replicas. For high DCs, three separate data placements were built. A total of 1,000 cloudlets are selected at random for the data replication order. The proposed MO-PSO and MO-ACO are evaluated against several well-known approaches. PSO, and ACO parameters settings are presented in TABLES 5 and 6 respectively.

TABLE 4. Simulation parameters of the configuration system.

No.	Entity proposed system	High datacenters	Mid datacenters	Low datacenters
	Datacenters (DCs)			
1	No. of DCs (21)	1	5	15
2	DCs Costs	500	300	100
3	No. of hosts per DC (110)	20	30	60
	Hosts			
4	Processing elements per host	12 - 16	4 - 8	1 - 4
5	Processing element MIPS	1000 - 2000	500 - 1000	100 - 500
6	Processing element bandwidth	5 – 10 GB	2 – 4 GB	1 – 2 GB
	Virtual Machines (VMs)			
7	No. of VMs (470)	200	150	120
8	MIPS	800	400	200
9	Memory	2 GB	1 GB	512 MB
10	Bandwidth	10 GB	2 GB	1 GB
11	No. of processing elements	8 - 16	4 - 8	1 - 4
12	Cloudlet scheduler	Time & space shard		
13	VM scheduler			
	Cloudlets			
14	Cloudlet tasks	1000 task		
15	Task length	1000 - 20000		
	Files			
	Using Zipf distribution and geometric distribution			
	In the Cloud storage, three separate data files are put, with a size within the [0.1, 10] GB range.			
16	No. of files (3)	A	B	C
17	Replication costs	500	300	100
	Users			
18	No. of users	10 – 50		

TABLE 5. PSO parameters.

No.	Parameters	Values
1	Number of particles	100
2	C_1	2
3	C_2	2
4	R_1	[0-1]
5	R_2	[0-1]
6	w_{max}	0.9
7	w_{min}	0.4
8	Number of iteration	1000
9	W	1
10	Population (Swarm size k)	50

TABLE 6. ACO parameters.

No.	Parameters	Values
1	α	1
2	β	2
3	p	0.3
4	q	1
5	m	100
6	t_k	1000
7	p_{τ_0}	0.9

TABLE 7. Related and the proposed work comparison.

Strategy	Year	Availability	Load balancing	Heterogeneity	Knapsack problem	Distance	Least cost path	Optimal location	Path Length Task	Replication decision	Multi-hop Tasks based on IOT *
SWORD [36]	2014	✓	✓								
D2RS [33]	2012	✓	✓								
MORM [58]	2013	✓	✓								
ADRS [35]	2015	✓	✓								
Fuzzy-FP [59]	2016	✓	✓								
PEPR [60]	2016	✓	✓								
EIMORM [61]	2017	✓	✓								
DPRS [32]	2017	✓	✓								
PDR [62]	2018	✓	✓								
EFS [63]	2011	✓	✓								
DCR2S [34]	2016	✓	✓	✓	✓						
ACO [64]	2016	✓	✓								
GASA [65]	2016	✓	✓								
DRSACO [66]	2013	✓	✓								
Genetic [67]	2015	✓	✓								
APSDRDO [68]	2018	✓	✓								
MOGA [48]	2019	✓	✓	✓							
Our Strategy	2020	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓*

In TABLE 7, a distinction is made between similar work and the proposed approach. It's clear that the suggested approach will handle a wide range of data replication and positioning problems.

VI. RESULTS AND ANALYSIS

CloudSim is used in this paper to run tests on the dynamic replica's collection and location in the cloud. Through comparing the experimental resultsthe proposed MO-PSO and MO-ACO algorithms are compared to sevderal well-known algorithms.

A. FIRST EXPERIMENTS FOR CLOUDLETS OPTIMIZATION

The MO-PSO and MO-ACO approaches were used to measure the selection and positioning of replicas using CloudSim. Cloudlets can be sent in CloudSim in 3 seconds, according to the findings of the experiments.The second experiment demonstrates that the suggested technique cuts CloudSim's overall execution time by 66% to less than one second. Experiments indicate that the suggested technique, which makes use of CloudSim, is the most effective method for picking and positioning replicas.

B. COMPUTATIONAL COMPLEXITY

The algorithm's efficiency is determined by its computational complexity, which is determined by the cost of running time using big-O algorithms [69]–[71]. The proposed MO-ACO and MO-PSO have $O(N^2)$, as shown in Algorithm 1 and Algorithm 2. The outer loop at most runs for N epochs, *i.e.*, No. of ants (*Nants*) times number of DCs (*No_DCs*).

The algorithm decides what is accepted and refused depend on the files put and takes into account the cost of file on the Dc in terms of cost complexity. As a result, the major three data file replicas must be mirrored on mid-to-low cost Dcs.

C. OPTIMAL REPLICA SELECTION EXPERIMENT

FIGURE 2 depicts the impact of utilizing the MO-PSO, EFS and DCR2S algorithms on replication costs as the number of tasks assigned to users grows. When opposed to DCR2S and EFS, it iis obvious that the suggested MO-PSO approach reduces replication costs. When evaluated against the DCR2S, whereas replication costs are equal to the budget, the EFS replication costs are high expensive. "Constant cost, maximize replica, and user waiting time" are the three limitations.The proposed MO-PSO is extremely useful for lowering replication costs, increasing availability, and lowering ITBDF. Cloudlets are utilized to optimize the replica selection process.

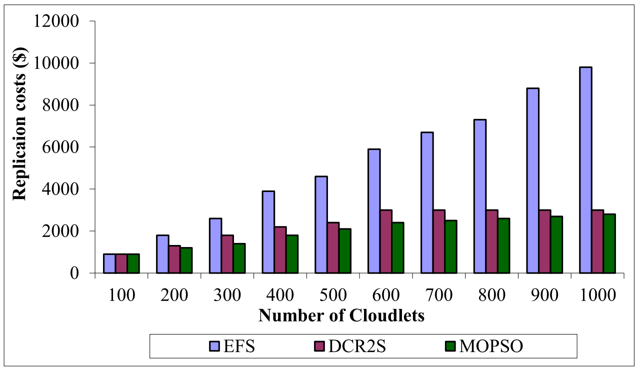


FIGURE 2. Replication costs with different number of cloudlets.

The simulation results in FIGURE 3 demonstrate the relationship between work time and job number by utilizing the proposed MO-PSO algorithm to reduce time costs. MO-PSO decreases execution time, replica access time, and the least-cost route. The simulation results revealed that the suggested technique outperforms ACO and GASA.

The simulation results demonstrate the MO-PSO algorithm's ability to pick optimal replicas depend on the ITBDF by accessing the most common replica. These results demonstrate that MO-PSO is more efficient than other algorithms in terms of the execution time required to access optimized replicas, as shown in FIGURE 4.

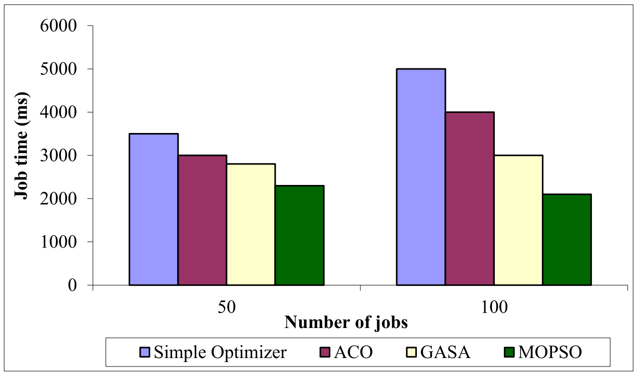


FIGURE 3. Average job time.

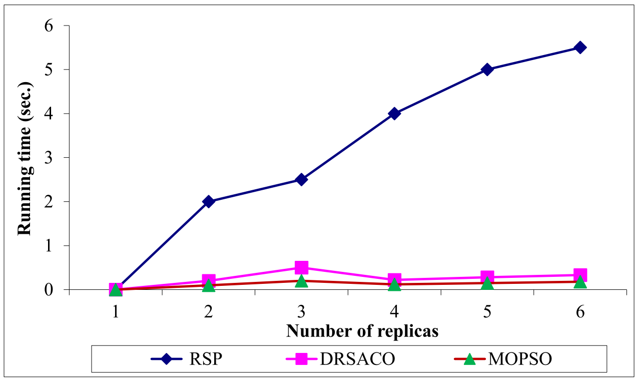


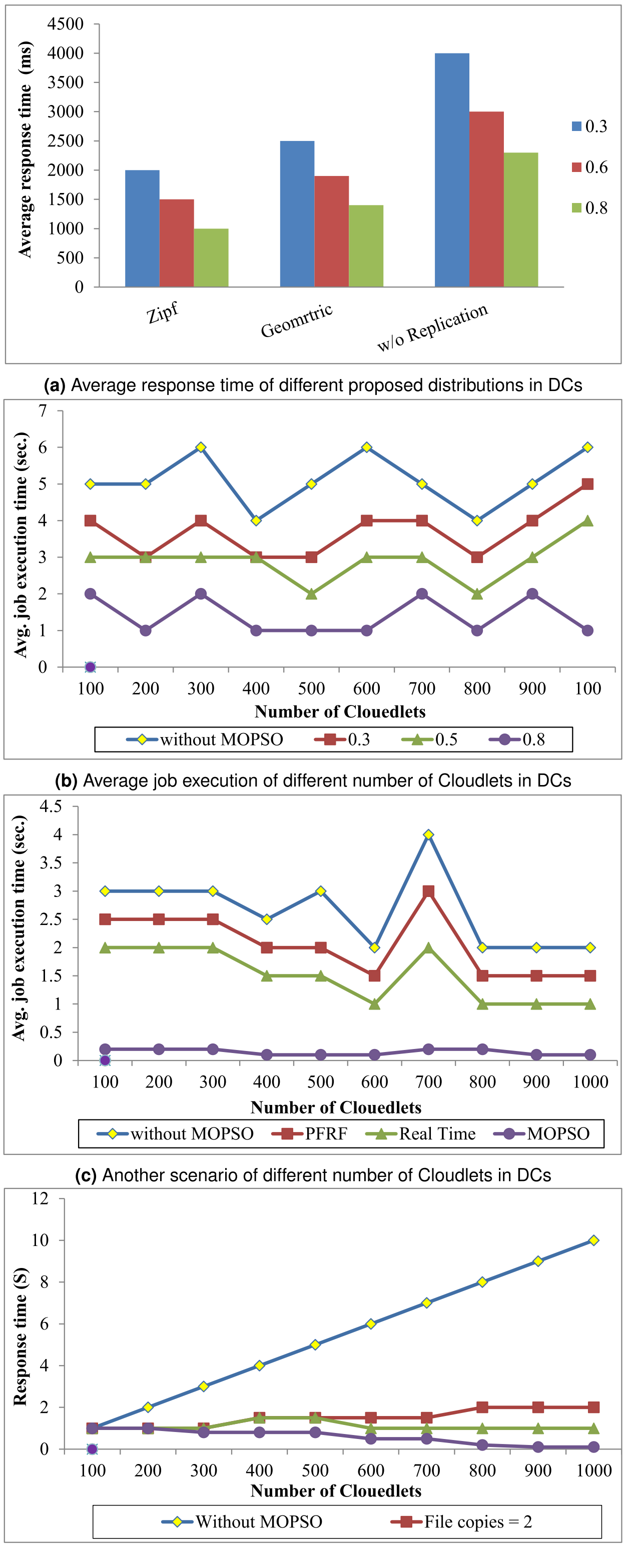
FIGURE 4. Running time for selecting optimal replica.

D. VARIOUS REPLICATION SCENARIOS USING ZIPF AND GEOMETRIC DISTRIBUTIONS

The computation of the Zipf and geometric distributions using MO-PSO is discussed in this section, in addition to a new technique for solving the optimal replica selection problem. FIGURE 5 shows a comparison of the MO-PSO algorithm with a real-time algorithm depend on Zipf and geometric distributions. As a result, the average response time for selecting data replicas grows as the number of data replicas grows. Inside the requested amount of data replicas, the MOPSO approach indicates a decreased average response time. The proposed MO-PSO solution is clearly surpassing to the other algorithms. FIGURE 5 depicts four selection replications in the cloud, each with a different average response time for data replication as a function of the number of replicas. From (a) to (d), a variety of scenarios are represented, and we've built various average response time scenarios ranging from 100 to 1000 cloudlets. When users need a certain number of data replicas, the MO-PSO technique will improve the average response time of choosing data replicas.

The below are the benefits and parameters of the proposed algorithm:

- Multi-Objective PSO (MO-PSO),
- Replication cost and time estimation,
- Fast replication speed with optimal nodes selection.



(a) Average response time of different proposed distributions in DCs

(b) Average job execution of different number of Cloudlets in DCs

(c) Another scenario of different number of Cloudlets in DCs

(d) Another scenario with selecting number of replicas

FIGURE 5. **Response time of different distributions scenarios in CloudSim.**

E. THE PROPOSED MO-ACO EXPERIMENT FOR REPLICAS PLACEMENT

The proposed MO-ACO algorithm is utilized to measure the location of replicas by ant behavior and the number of cycles

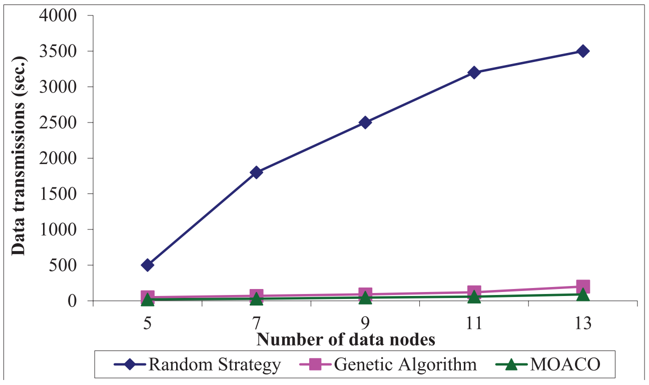


FIGURE 6. **Data transmission with different numbers data nodes.**

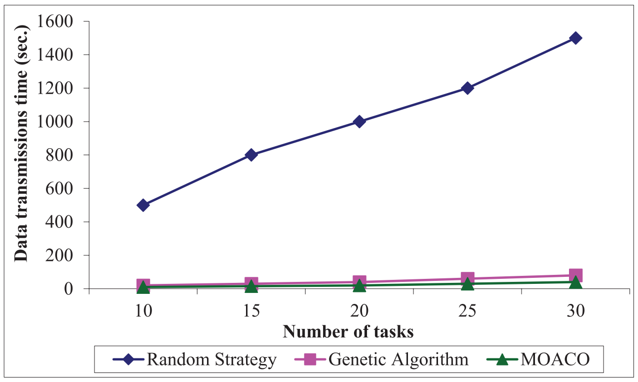


FIGURE 7. **Data transmission with different numbers of tasks.**

by DCs in this experiment. In CloudSim, various distributions utilized including Zipf and geometric distributions to determine the best placements in the DCs. As opposed to other algorithms, the tests reveal that our MO-ACO approach produces the best outcomes. The proposed approach is assessed depend on a number of factors, including the least-cost direction and distance to the best replica placement positions in DCs using Zipf and geometric distributions. Depending on the number of tasks and nodes, different methods are utilized to assess the transmission mechanism in DCs. In FIGURE 6 to FIGURE 8, MO-ACO is utilized to measure the amount of replicas distributed over DCs in addition to the data transfer rate, which saves time and money. The results demonstrate that the proposed algorithm outperforms the other algorithms.

F. SHORTEST PATH SELECTION USING PROPOSED MO-ACO

An estimation of the shortest path using MO-ACO is seen in this part, in addition to a novel method for solving optimal placement. FIGURE 9 provides a comparison of the MO-ACO algorithm versus the GASA algorithm, all of which are depend on the shortest path issue and influence the optimum data replica's shortest path in nodes.

The data replications' shortest path grows exponentially as the number of data replicas grows. Within the number of data replications requested, the suggested MO-ACO technique

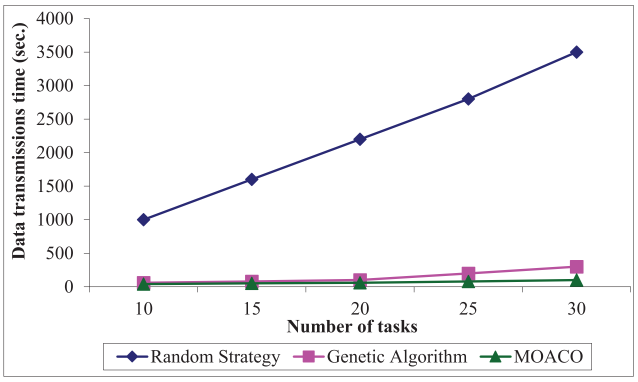


FIGURE 8. Total data transmission with different number of tasks.

decreases the shortest path. When comparing the MOACO technique to the GASA technique in terms of minimizing the shortest path, it is obvious that the MO-ACO technique is superior. FIGURE 9 depicts many scenarios for the expense of data duplication based on the number of replicas for four of the shortest path cases. There are several scenarios ranging from (a) to (c), and we've built various shortest path scenarios using 1 to 13 data nodes and 1000 cloudlets. When consumers need a large number of data replicas, the MOACO technique will improve data file replication's shortest path.

The advantages and specifications of the proposed algorithm are listed as follows:

- Multi-Objective ACO (MO-ACO)algorithm,
- Replication cost and time estimation,
- Fast replication speed with optimal nodes placement.
- Selecting the shortest path.

G. REPLICATION COST AND KNAPSACK PROBLEM

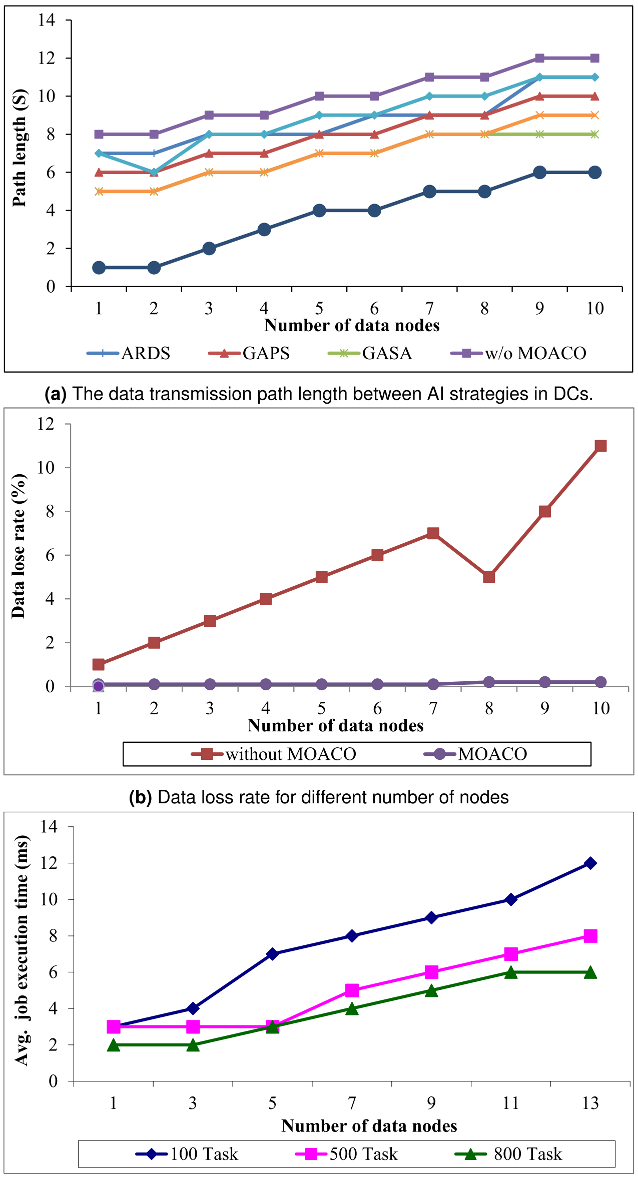
FIGURE 10 depicts a trade-off between the MOACO and DCR2S algorithms depend on the replication budget with the knapsack dilemma, which has an effect on the data replication budget. As a result, as the number of data replicas increases, so do the costs of data replication. Based on the number of replicas needed by users, the proposed MOACO algorithm has a low cost. It's easy to see how the MOACO technique outperforms the DCR2S algorithm. FIGURE 10 depicts a variety of expense scenarios for data duplication. It is obvious from (a) to (d) that the suggested MO-ACO technique can easily improve the file replication budget.

VII. PERFORMANCE EVALUATION

The proposed models were evaluated using five performance indicators. Average response time, efficient network use, capacity usage, replication frequency, and impact ratio are the requirements.

A. AVERAGE RESPONSE TIME

In FIGURE 11, the replication average response time using Zipf and geometric distributions are shown. From simulation results, one can notice that the suggested technique



(a) The data transmission path length between AI strategies in DCs.

(b) Data loss rate for different number of nodes

(c) Average job execution time for different number of nodes

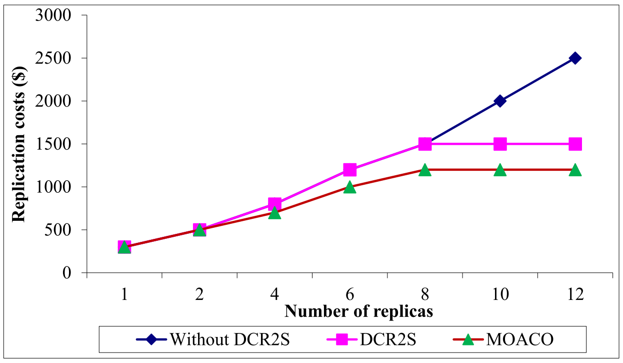
FIGURE 9. Different scenarios of replications over number of nodes.

minimizes the average response time by 12% compared with the well-known PDR approach. If the number of user activities for collection and placement replications grows, the total response time increases exponentially, but our approach decreases time effectively. These data can be estimated using Eq. (24) as follows [32]:

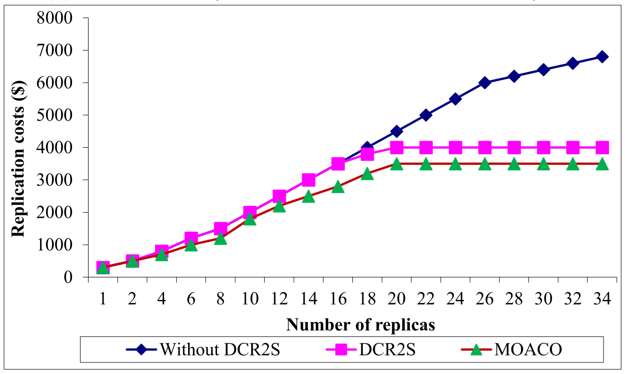
$$ART = \sum_{i=1}^{n} \sum_{j=1}^{mi} (js_{ij}(rt) - js_{ij}(st)) / \sum_{i=1}^{m} mi \tag{24}$$

where

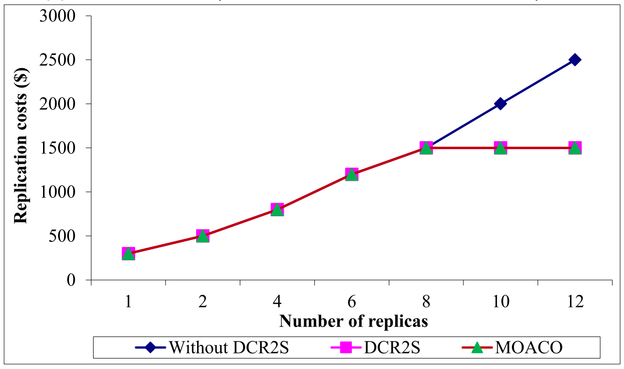
ART	Average Response Time
$js_{ij}(rt)$	receiving time
$js_{ij}(st)$	sending time
mi	number of tasks



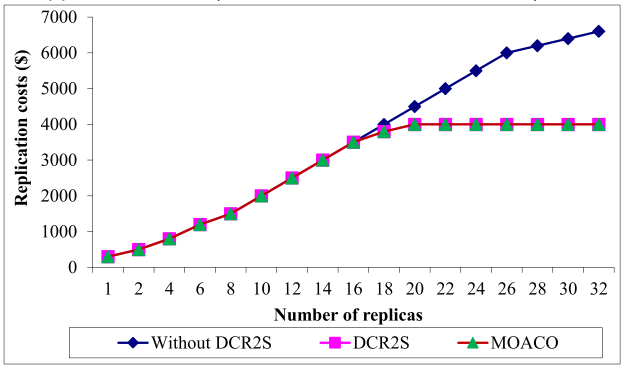
(a) 1st scenario of replication cost with different number of replicas



(b) 2nd scenario of replication cost with different number of replicas



(c) 3rd scenario of replication cost with different number of replicas



(d) 4th scenario of replication cost with different number of replicas

FIGURE 10. Different scenarios of data replication cost with the number of replicas according budget.

B. EFFECTIVE NETWORK USAGE (ENU)

In FIGURE 12, effective network utilization tests the percentage of data passing across the domain network on a scale from 0 to 1. Use requires the number of appropriate accesses, reaction time and replications to other locations that are nearby and have less expensive between DCs. The use indicates that the bandwidth usage was more stable and consistent for the proposed approach. (25) as follows [32]:

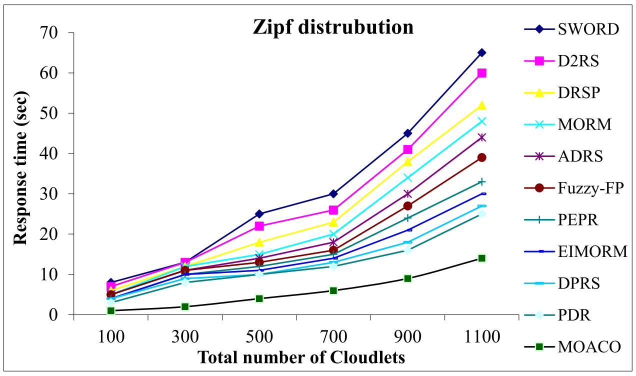


FIGURE 11. Average response time for various data replication techniques.

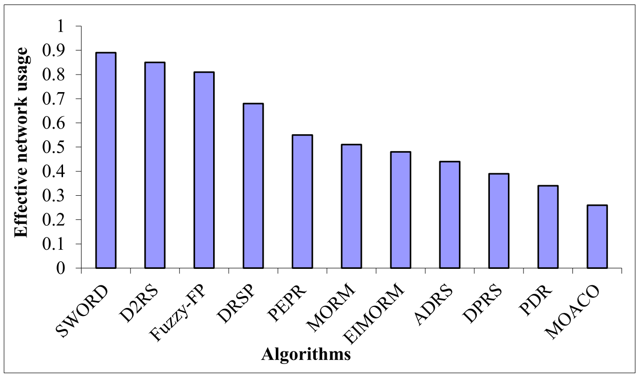


FIGURE 12. Effective network usage for different data replication techniques.

$$ENU = N_{rfa} + N_{fa}/N_{lfa} \tag{25}$$

where

ENU	Effective Network Usage
N_{rfa}	number of access file from faraway site
N_{fa}	total time of replication operation
N_{lfa}	number of access file from near site

C. STORAGE USAGE

In FIGURE 13, the various storage consumption rates in proposed scheme determine the relative data size when transmitting and storing data replicas with respect to the DC size. However, the rate is another indicator of the costs and the time of transition between the DCs. (26) as follows [32]:

$$SU = FileSpaceAvailable/Space \tag{26}$$

D. REPLICATION FREQUENCY

In FIGURE 14, the level of replication tests users' access to data by increasing the number of replicas. In these cases,

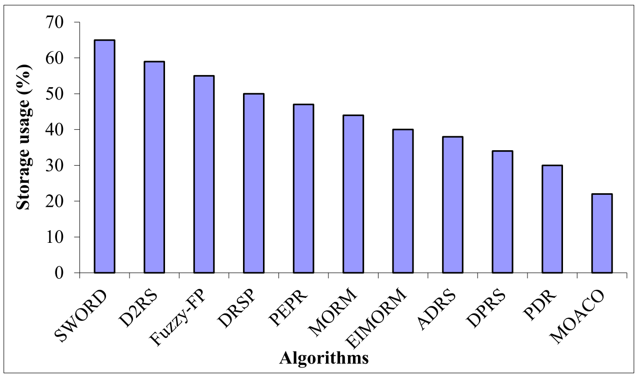


FIGURE 13. Storage usage rates for different data replication techniques.

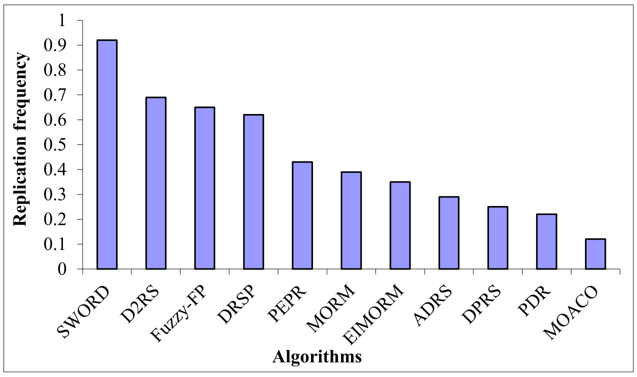


FIGURE 14. Replication frequencies for different data replication techniques.

a higher replication level is achieved and load congestion between DCs, the network, etc. is improved. However, our strategy has access to the most common data with high replica placement percentages among DCs. The experiments have shown that the proposed approach is outperforms the most common PDR algorithm based on availability and minimizing DC congestion.

E. HIT RATIO

FIGURE 15, the hit ratio is defined as the proportion of available data that is either nearer to or further away from tasks at distances from 1000 to 3000. Our approach has obtained a higher hit ratio relative to the most common PDR technique, which reveals that our system can be adapted precisely to the actions of model users when accessing nearer and further replications. Hit ratio is presented in Eq. (27) [32]:

$$HR = \text{No. of local file access/ No. of local file access} + \text{No. of replicas} + \text{No. of remote file access} \tag{27}$$

F. DISCUSSIONS

The experiments discussed can be outlined as follows: First, the proposed MO-PSO technique is utilized to access the most common data using user request tasks to access data replicas in the least period of time for the lowest cost and

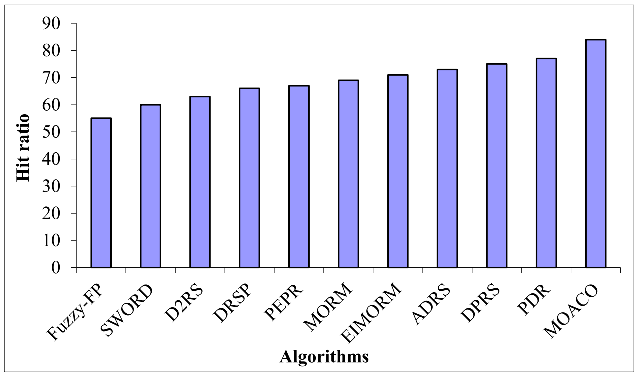


FIGURE 15. Hit Ratios for various data replication techniques.

maximum availability after several attempts according to the PSO requirements. Second, the proposed MO-ACO technique is utilized to position or place replications at appropriate locations based on prices, time and distances between DCs. In addition, global search is improved by discovering a new search field. The suggested technique produces results in any period by constructing ACO solutions for expense, availability, distance and time of data replication. The best location of data duplication will be maintained as the best global approach following many attempts to use the criterion.

VIII. CONCLUSION AND FUTURE WORK

The cloud ecosystem is one of the most critical scenarios for achieving high availability and maximize replica efficiency. This paper introduced two bio-based algorithms (MOPSO and MO-ACO) for dynamic data replication and positioning. MO-PSO provides optimal access to the most common data replicas and optimizes the replication using ITBDF. The proposed MO-ACO algorithm is utilized to refine the positioning of data replicas previously identified using MO-PSO algorithm in appropriate locations near users. The proposed system architecture was designed and executed using CloudSim. The efficiency of the suggested technique was evaluated versus various replication techniques like ADRS, D2RS, DRACO, EFS, and GA. The results of the simulation indicated that the algorithms proposed were more effective and superior than the algorithms compared.

In future work, the proposed system architecture will be tested using on a real cloud computing environment. Moreover, we're going to consider two interesting issues about replica collection and positioning. First, the accuracy and energy efficiency of complex data replication. Second, latest swarm intelligence dynamic data replication algorithms in cloud computing will be used.

REFERENCES

[1] R. Salem, M. A. Salam, H. Abdelkader, and A. A. Mohamed, "An artificial bee colony algorithm for data replication optimization in cloud environments," *IEEE Access*, vol. 8, pp. 51841–51852, 2020.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[4] S. Long and Y. Zhao, "A toolkit for modeling and simulating cloud data storage: An extension to CloudSim," in *Proc. Int. Conf. Control Eng. Commun. Technol.*, Dec. 2012, pp. 597–600.

[5] Y. Mansouri and R. Buyya, "To move or not to move: Cost optimization in a dual cloud-based storage architecture," *J. Netw. Comput. Appl.*, vol. 75, pp. 223–235, Nov. 2016.

[6] N. Rajeshirke, R. Sawant, S. Sawant, and H. Shaikh, "Load balancing in cloud computing," *Int. J. Recent Trends Eng. Res.*, vol. 3, no. 3, pp. 260–267, 2017.

[7] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," *J. Netw. Comput. Appl.*, vol. 71, pp. 86–98, Aug. 2016.

[8] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 50–71, Jun. 2017.

[9] R. K. Naha and M. Othman, "Cost-aware service brokering and performance sentient load balancing algorithms in the cloud," *J. Netw. Comput. Appl.*, vol. 75, pp. 47–57, Nov. 2016.

[10] H.-Y. Ahn, K.-H. Lee, and Y.-J. Lee, "Dynamic erasure coding decision for modern block-oriented distributed storage systems," *J. Supercomput.*, vol. 72, no. 4, pp. 1312–1341, Apr. 2016.

[11] T. Zhu, D. Feng, F. Wang, Y. Hua, Q. Shi, Y. Xie, and Y. Wan, "A congestion-aware and robust multicast protocol in SDN-based data center networks," *J. Netw. Comput. Appl.*, vol. 95, pp. 105–117, Oct. 2017.

[12] H. F. El Yamany, M. F. Mohamed, K. Grolinger, and M. A. M. Capretz, "A generalized service replication process in distributed environments," in *Proc. 5th Int. Conf. Cloud Comput. Services Sci.*, vol. 1, 2015, pp. 186–193.

[13] T. Rashmi Ranjana, D. Jayalakshmi, and R. Srinivasan, "On replication strategies for data intensive cloud applications," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 3, pp. 2479–2484, 2017.

[14] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Comput.*, vol. 18, no. 1, pp. 385–402, Mar. 2015.

[15] P. J. Kumar, V. University, and P. Ilango, "Data replication in current generation computing environment," *Int. J. Eng. Trends Technol.*, vol. 45, no. 10, pp. 488–492, Mar. 2017.

[16] T. Chen, R. Bahsoon, and A.-R.-H. Tawil, "Scalable service-oriented replication with flexible consistency guarantee in the cloud," *Inf. Sci.*, vol. 264, pp. 349–370, Apr. 2014.

[17] N. Maheshwari, R. Nanduri, and V. Varma, "Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 119–127, Jan. 2012.

[18] B. Alami Milani and N. Jafari Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions," *J. Netw. Comput. Appl.*, vol. 64, pp. 229–238, Apr. 2016.

[19] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1002–1026, 2nd Quart., 2017.

[20] P. Shrivastava, U. Chourasia, and S. Agrawal, "A survey on datacenter placement and replication approach for cloud," *Int. J. Comput. Appl.*, vol. 141, no. 10, pp. 43–45, May 2016.

[21] R. R. Karandikar and M. Gudadhe, "Comparative analysis of dynamic replication strategies in cloud," *Int. J. Comput. Appl.*, vol. 975, no. 1, pp. 26–32, 2016.

[22] S. Dayyani and M. R. Khayyambashi, "A comparative study of replication techniques in grid computing systems," *Int. J. Comput. Sci. Inf. Secur.*, vol. 11, no. 9, pp. 64–73, Sep. 2013.

[23] T. Hamrouni, S. Slimani, and F. B. Charrada, "A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids," *Eng. Appl. Artif. Intell.*, vol. 48, pp. 140–158, Feb. 2016.

[24] S. Bansal, S. Sharma, and I. Trivedi, "A dynamic replica placement algorithm to enhance multiple failures capability in distributed system," *Int. J. Distrib. Parallel Syst.*, vol. 2, no. 6, p. 79, 2011.

[25] Z. Fadaie and A. M. Rahmani, "A new replica placement algorithm in data grid," *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, p. 491, 2012.

[26] S. Limam, R. Mokadem, and G. Belalem, "Data replication strategy with satisfaction of availability, performance and tenant budget requirements," *Cluster Comput.*, vol. 22, no. 4, pp. 1199–1210, Dec. 2019.

[27] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 705–718, Jul. 2019.

[28] A. Rajalakshmi, D. Vijayakumar, and K. Srinivasagan, "An improved dynamic data replica selection and placement in hybrid cloud," *Netw. Commun. Eng.*, vol. 6, no. 3, pp. 112–117, 2014.

[29] G. Liu, H. Shen, and H. Chandler, "Selective data replication for online social networks with distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2377–2393, Aug. 2016.

[30] A. Rama and M. Reddy, "Data replication system in cloud based on data mining techniques," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 4216–4221, 2013.

[31] T. Loukopoulos and I. Ahmad, "Static and adaptive distributed data replication using genetic algorithms," *J. Parallel Distrib. Comput.*, vol. 64, no. 11, pp. 1270–1285, Nov. 2004.

[32] N. Mansouri, M. K. Rafsanjani, and M. M. Javidi, "DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments," *Simul. Model. Pract. Theory*, vol. 77, pp. 177–196, Sep. 2017.

[33] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, and X.-W. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments," *J. Comput. Sci. Technol.*, vol. 27, no. 2, pp. 256–272, Mar. 2012.

[34] N. K. Gill and S. Singh, "A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers," *Future Gener. Comput. Syst.*, vol. 65, pp. 10–32, Dec. 2016.

[35] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement," *Frontiers Comput. Sci.*, vol. 10, no. 5, pp. 925–935, Oct. 2016.

[36] K. A. Kumar, A. Quamar, A. Deshpande, and S. Khuller, "SWORD: Workload-aware data placement and replica selection for cloud data management systems," *VLDB J.*, vol. 23, no. 6, pp. 845–870, Dec. 2014.

[37] X. Bai, H. Jin, X. Liao, X. Shi, and Z. Shao, "RTRM: A response time-based replica management strategy for cloud storage system," in *Proc. Int. Conf. Grid Pervas. Comput.* Cham, Switzerland: Springer, 2013, pp. 124–133.

[38] D. Yuan, Y. Yang, X. Liu, and J. Chen, "A data placement strategy in scientific cloud workflows," *Future Gener. Comput. Syst.*, vol. 26, no. 8, pp. 1200–1214, Oct. 2010.

[39] C. Hamdeni, T. Hamrouni, and F. B. Charrada, "Adaptive measurement method for data popularity in distributed systems," *Cluster Comput.*, vol. 19, no. 4, pp. 1801–1818, Dec. 2016.

[40] Q. Xu, Z. Xu, and T. Wang, "A data-placement strategy based on genetic algorithm in cloud computing," *Int. J. Intell. Sci.*, vol. 5, no. 3, pp. 145–157, 2015.

[41] Z. Li, W. Cai, and S. J. Turner, "Un-identical federate replication structure for improving performance of HLA-based simulations," *Simul. Model. Pract. Theory*, vol. 48, pp. 112–128, Nov. 2014.

[42] Z. Tang, X. Zhang, K. Li, and K. Li, "An intermediate data placement algorithm for load balancing in spark computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 287–301, Jan. 2018.

[43] T.-Y. Hsu, A. D. Kshemkalyani, and M. Shen, "Causal consistency algorithms for partially replicated and fully replicated systems," *Future Gener. Comput. Syst.*, vol. 86, pp. 1118–1133, Sep. 2018.

[44] H. Casanova, Y. Robert, F. Vivien, and D. Zaidouni, "On the impact of process replication on executions of large-scale parallel applications with coordinated checkpointing," *Future Gener. Comput. Syst.*, vol. 51, pp. 7–19, Oct. 2015.

[45] P. Matri, M. S. Pérez, A. Costan, L. Bougé, and G. Antoniu, "Keeping up with storage: Decentralized, write-enabled dynamic geo-replication," *Future Gener. Comput. Syst.*, vol. 86, pp. 1093–1105, Sep. 2018.

[46] M.-C. Lee, F.-Y. Leu, and Y.-P. Chen, "PFRF: An adaptive data replication algorithm based on star-topology data grids," *Future Gener. Comput. Syst.*, vol. 28, no. 7, pp. 1045–1057, Jul. 2012.

[47] N. Mansouri, B. Mohammad Hasani Zade, and M. M. Javidi, "A multi-objective optimized replication using fuzzy based self-defense algorithm for cloud computing," *J. Netw. Comput. Appl.*, vol. 171, Dec. 2020, Art. no. 102811.

[48] C. Li, Y. Wang, H. Tang, and Y. Luo, "Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud," *Future Gener. Comput. Syst.*, vol. 100, pp. 921–937, Nov. 2019.

[49] 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 Comput., Pract. Exper.*, vol. 31, no. 1, p. e4757, Jan. 2019.

[50] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems," *Int. J. Electr. Power Energy Syst.*, vol. 34, no. 1, pp. 66–74, Jan. 2012.
[51] M. Vidhya and N. Sadhasivam, "Parallel particle swarm optimization for reducing data redundancy in heterogeneous cloud storage," *IJTET*, vol. 3, no. 1, pp. 73–78, 2015.
[52] K. Kavitha and R. Neela, "Optimal allocation of multi-type FACTS devices and its effect in enhancing system security using BBO, WIPSO & PSO," *J. Electr. Syst. Inf. Technol.*, vol. 5, no. 3, pp. 777–793, Dec. 2018.
[53] Y. Moon, H. Yu, J.-M. Gil, and J. Lim, "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments," *Hum.-Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 28, Dec. 2017.
[54] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 321–332, Aug. 2002.
[55] R. Gao and J. Wu, "Dynamic load balancing strategy for cloud computing with ant colony optimization," *Future Internet*, vol. 7, no. 4, pp. 465–483, Nov. 2015.
[56] J. Lin, L. Yang, and Y. Zheng, "A replica selection strategy on ant-algorithm in data-intensive applications," *Int. J. Online Biomed. Eng.*, vol. 9, no. S6, p. 38, Jun. 2013.
[57] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, Dec. 2013.
[58] S.-Q. Long, Y.-L. Zhao, and W. Chen, "MORM: A multi-objective optimized replication management strategy for cloud storage cluster," *J. Syst. Archit.*, vol. 60, no. 2, pp. 234–244, Feb. 2014.
[59] P. Elango, G. Arts, S. College, and K. Samy, "Fuzzy FP-tree based data replication management system in cloud," *Int. J. Eng. Trends Technol.*, vol. 36, no. 9, pp. 481–489, May 2016.
[60] U. Tos, R. Mokadem, A. Hameurlain, T. Ayav, and S. Bora, "A performance and profit oriented data replication strategy for cloud systems," in *Proc. Int. IEEE Conferences Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr.*, Jul. 2016, pp. 780–787.
[61] E. B. Edwin, P. Umamaheswari, and M. R. Thanka, "An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center," *Cluster Comput.*, vol. 22, no. S5, pp. 11119–11128, Sep. 2019.
[62] N. Mansouri and M. M. Javidi, "A new prefetching-aware data replication to decrease access latency in cloud environment," *J. Syst. Softw.*, vol. 144, pp. 197–215, Oct. 2018.
[63] M. Bsoul, A. Al-Khasawneh, E. E. Abdallah, and Y. Kilani, "Enhanced fast spread replication strategy for data grid," *J. Netw. Comput. Appl.*, vol. 34, no. 2, pp. 575–580, Mar. 2011.
[64] N. J. Navimipour and B. A. Milani, "Replica selection in the cloud environments using an ant colony algorithm," in *Proc. 3rd Int. Conf. Digit. Inf. Process., Data Mining, Wireless Commun. (DIPDMWC)*, Jul. 2016, pp. 105–110.
[65] T. Junfeng and L. Weiping, "Pheromone-based genetic algorithm adaptive selection algorithm in cloud storage," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 6, pp. 269–278, Jun. 2016.
[66] L. Wang, J. Luo, J. Shen, and F. Dong, "Cost and time aware ant colony algorithm for data replica in alpha magnetic spectrometer experiment," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2013, pp. 247–254.
[67] P. Jayasree and V. Saravanan, "Apsdrdo: Adaptive particle swarm division and replication of data optimization for security in cloud computing," *IOSR J. Eng.*, vol. 1, no. 1, pp. 1–8, 2018.
[68] L. Cui, J. Zhang, L. Yue, Y. Shi, H. Li, and D. Yuan, "A genetic algorithm based data replica placement strategy for scientific applications in clouds," *IEEE Trans. Services Comput.*, vol. 11, no. 4, pp. 727–739, Jul. 2018.
[69] S. Qaiyum, I. Aziz, M. H. Hasan, A. I. Khan, and A. Almalawi, "Incremental interval type-2 fuzzy clustering of data streams using single pass method," *Sensors*, vol. 20, no. 11, p. 3210, Jun. 2020.
[70] N. Bharanidharan and H. Rajaguru, "Performance enhancement of swarm intelligence techniques in dementia classification using dragonfly-based hybrid algorithms," *Int. J. Imag. Syst. Technol.*, vol. 30, no. 1, pp. 57–74, Mar. 2020.
[71] X. Liu, B. G. Evans, and K. Moessner, "Energy-efficient sensor scheduling algorithm in cognitive radio networks employing heterogeneous sensors," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1243–1249, Mar. 2015.

AHMED AWAD is currently a Ph.D. Researcher with Menoufia University. He is also an Assistant Lecturer with the Cairo Higher Institute for Languages and Administrative Science. His major research interests include data replication and cloud computing. He also registered a four patent in Egyptian patent office.

RASHED SALEM received the B.Sc. degree in information systems and the M.Sc. degree, in 2002 and 2007, respectively, and the Ph.D. degree in informatics from the University of Lyon 2, France, in 2012. He is currently an Associate Professor with the Information Systems Department, Faculty of Computers and Information, Menoufia University, Egypt. He has participated more than 30 technical articles. His current research interests include distributed databases, data warehousing and business intelligence (BI), machine learning, data mining, and big data analytics.

HATEM ABDELKADER received the Ph.D. degree from the Faculty of Engineering, Alexandria University, Egypt, in 2001. He has been a Professor with the Information Systems Department, Faculty of Computers and Information, Menoufia University, Egypt, since 2004. He has participated in more than 150 technical articles in the areas of ANNs, database management systems, and data mining.

MUSTAFA ABDUL SALAM received the Ph.D. degree in information systems from the Faculty of Computers and Information, Cairo University, Egypt. He is currently an Assistant Professor with the Artificial Intelligence Department, Faculty of Computers and Artificial Intelligence, Benha University, Egypt. He has contributed more than 25 technical articles in the areas of ANN, SVM, swarm intelligence optimization, time series forecasting, extreme learning machine, and hybrid AI models in international journals and conferences. His majors research interests include machine learning, deep learning, big data analytic, and stream data mining.